

Travail de bachelor 2010

Filière Informatique de gestion

Outschool, plateforme Web pour camps scolaires



Etudiant-e : Barnabé Pache

Professeur : Florian Doche

Préface

De nos jours, de plus en plus de start-ups voient le jour. Certaines institutions comme la "Business Expérience" de la HES-SO Valais, proposent d'aider les étudiants à monter leur propre entreprise. Cette dernière les aide d'un point de vue financier mais également au niveau du suivi du travail avec des personnes très compétentes qui sont à la disposition des nouveaux entrepreneurs. De plus en plus de start-ups, proposent leurs produits directement sur Internet. Ce moyen de présenter ses produits permet d'avoir beaucoup plus de clients potentiels. Mais une partie très importante de ce système est la sécurité.

En cas de faille de sécurité sur une plateforme Web, tous les avantages d'une entreprise basée sur le Web peuvent se transformer en inconvénients et ainsi forcer la fermeture de cette dernière. C'est pourquoi l'aspect sécurité est très important sur Internet.

Résumé

Le projet outhschool.ch est une start-up aidée par "Business Expérience" qui propose aux différentes personnes tel que des professeurs de les aider à organiser des camps scolaires. En date du 17 mai 2010, l'entreprise outhschool ne dispose pas encore de plateforme Web. Toutes demandes faites par des personnes désirant organiser des camps scolaires, sont traitées de manière manuelle.

L'idée de ce projet et d'informatiser au maximum le processus de réservation afin de diminuer la masse de travail manuel de l'équipe outhschool. Une version 1.0 a été mise en production au début du mois de juin 2010. Mais cette version ne fait que de proposer un formulaire qui envoie un courriel à l'équipe d'outhschool avec les besoins nécessaires. Le traitement se fait ensuite de manière manuelle. La version 1.1 qui est développée dans le cadre de ce travail de bachelor offre les différentes fonctionnalités :

- ❑ Possibilité à une personne désirant mettre à disposition des utilisateurs de la plateforme faisant une réservation un bien ou un service.
- ❑ Pour ces personnes-là, un outil de gestion de leurs biens ou services sera également mise en place sous forme de gadget Windows.
- ❑ Un formulaire dynamique de réservation donnera la possibilité à l'utilisateur de choisir lui-même les biens ou les services désirés parmi une liste affichée suivant différents paramètres.
- ❑ Toutes les réservations, composées de différents biens ou services ainsi que les coordonnées de contact, seront enregistrées dans une base de données.
- ❑ Cette plateforme permettra aussi l'intégration des biens ou de services depuis des bases de données externes.

Structure du document

Ci-dessous se trouve une table des matières qui permet d'avoir un aperçu de toutes les différentes sections de ce document.

La section "**Introduction**" permet de bien comprendre le contexte du projet. Les différents objectifs de ce projet sont également diffusés dans cette partie ainsi que les motivations personnelles.

La section "**Etat de l'art**" permet de comprendre les différents choix technologiques qui ont été fait dans le cadre de ce projet.

La section "**Le projet outhschool.ch**" explique de manière détaillée le fonctionnement de l'application ainsi que les fonctionnalités déjà existantes. Un aperçu de l'architecture et du diagramme de classes fait également partie de cette section.

Les sections "**Développement de la plateforme Web**" et "**Développement du gadget Windows**" expliquent, à l'aide de captures d'écran de l'application, le fonctionnement général de la plateforme. Les détails techniques de la mise en place de ces applications sont également détaillés à cet endroit.

La section "**Problèmes rencontrés**" liste tous les problèmes ainsi que leurs solutions qui ont été rencontrés dans le cadre de ce projet.

La section "**Améliorations**" permet d'avoir un aperçu des possibilités de développement futur de cette plateforme.

La section "**Gestion de projet**" permet de prendre connaissance des fonctionnalités réalisées ou pas par rapport au cahier des charges fixé en début de projet. Une explication générale sur le déroulement du projet est également faite dans cette partie.

La section "**Conclusion**" tire un bilan final du projet mais également un bilan personnel de ce que m'a apporté cet exercice.

Remerciements

Je tiens à remercier les différentes personnes qui m'ont aidé et suivi dans l'accomplissement de ce travail. Tout d'abord, Monsieur Florian Doche, professeur responsable du projet, qui a été disponible et à mon écoute durant tout ce projet. Il m'a également donné beaucoup de conseils au niveau de l'ergonomie des applications.

D'autre part, je tiens à remercier Monsieur Jean-Blaise Rey, développeur de la version 1.0 de la plateforme et qui m'a persuadé d'utiliser des technologies que je ne connaissais pas du tout.

Je remercie également toute l'équipe d'outhschool, composée de Messieurs Arnaud Cotter, Benoit Goncerut et Nicolas Ansaldo, pour la confiance qu'ils m'ont accordée et la rapidité de leurs réponses.

Finalement, je remercie les membres de ma famille qui ont relu ce document et qui m'ont aidé à le structurer correctement.

Déclaration sur l'honneur

"Je déclare, par ce document, que j'ai effectué le travail de bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail."

Pache Barnabé

Table des matières

1	Introduction	1
1.1	Contexte général	1
1.2	Descriptif du projet.....	1
1.3	Motivations personnelles	2
1.4	Objectifs.....	3
2	Etat de l'art	4
2.1	Les langages de programmation Web	4
2.1.1	Introduction	4
2.1.2	Les langages de programmation Web dynamiques	5
2.1.3	Résultats.....	7
2.1.4	Décisions prises pour le projet	8
2.2	Les Web services	8
2.2.1	Introduction	8
2.2.2	Les outils existant pour le Framework Django	10
2.2.3	Les langages d'échange de données.....	10
2.2.4	Résultats.....	12
2.2.5	Décisions prises pour le projet	12
3	Le projet outhschool.ch	13
3.1	Description détaillée du projet	13
3.2	Fonctionnalités déjà existantes	14
3.3	Architecture	15
3.4	Diagramme de classes	16
4	Développement de la plateforme Web.....	17
4.1	Fonctionnalités	17
4.1.1	Recherche multicritère de packages	17
4.1.2	Formulaire d'inscription pour les partenaires et prestataires	17
4.1.3	Formulaire de réservation	20
4.1.4	Zone d'administration	24
4.1.5	Le Web service.....	24
4.2	Technologies utilisées	25
4.2.1	Framework Django	25
4.2.2	Environnement de développement.....	26
4.2.3	SGBD.....	27
4.2.4	Serveur Web.....	27
5	Développement du gadget Windows 7	28
5.1	Qu'est-ce qu'un gadget ?	28
5.2	Fonctionnalités	28
5.2.1	Vérification de l'état du serveur	28
5.2.2	Authentification	29
5.2.3	Visualisation des prochaines réservations	29
5.2.4	Affichage dynamique de nouvelles réservations	29
5.2.5	Gestion des réservations.....	30
5.2.6	Gestion des biens/services	31
5.3	Ajouter / modifier un bien ou un service	31
5.4	Gestion des disponibilités.....	32
5.5	Technologies utilisées	33
5.5.1	HTML et CSS	33

5.5.2	Javascript avec l'objet XMLHttpRequest.....	34
5.5.3	Librairie JQuery	35
6	Problèmes rencontrés.....	36
6.1	Délai du cahier des charges.....	36
6.2	Nouvelles technologies	36
6.2.1	Nouveau langage	36
6.2.2	Nouvel environnement de développement.....	36
6.2.3	API pour les requêtes SQL.....	36
6.2.4	Configuration de l'envoi d'email.....	36
6.2.5	Récupération du projet existant.....	36
6.3	Communication avec groups.ch	37
6.4	Communication entre les applications.....	37
7	Améliorations.....	38
8	Gestion de projet	39
8.1	Technique de gestion de projet	39
8.2	Fonctionnalités réalisées.....	39
8.3	Ecart de temps entre prévu et réalisé	40
8.4	Déroulement du projet.....	40
8.5	Nombre d'heures réalisées	41
9	Conclusion	42
9.1	Généralités sur le projet	42
9.2	D'un point de vue personnel.....	42
10	Tables des illustrations.....	43
10.1	Table des schémas	43
10.2	Table des extraits de code	43
10.3	Table des écrans	43
11	Sources.....	44
12	Annexes.....	I
12.1	Cahier des charges	I
12.1.1	Architecture	I
12.1.2	Fonctionnalités sur l'existant	I
12.1.3	Fonctionnalités pour l'équipe outhschool.ch	I
12.1.4	Fonctionnalités concernant les prestataires de services.....	II
12.1.5	Fonctionnalités optionnelles	II
12.2	Diagramme de Use-Case	III
12.3	Comment reprendre le projet	V
12.3.1	Installer Python	V
12.3.2	Installer django	V
12.3.3	Installer les composants nécessaires.....	V
12.3.4	Installer MySQL	V
12.3.5	Installer la base de données	V
12.3.6	Installation du plugin eclipse "Py-Dev"	VI
12.3.7	Ouvrir le projet Django.....	VI
12.4	Contenu du CD-ROM	VII
12.5	Détail des heures de travail.....	VII

1 Introduction

1.1 Contexte général

Ce travail a été réalisé dans le cadre de la formation d'informaticien de gestion à la HES-SO Valais/Wallis pour l'obtention du bachelor en informatique de gestion. Il doit être réalisé en 360 heures minimum entre le 17 mai 2010 et le 16 août 2010. Étant donné que le 6^{ème} semestre de la formation s'est terminé le 2 juillet 2010, l'activité des 6 premières semaines s'élève à environ 50%.

Après avoir choisi six sujets, exposés par ordre de préférence, parmi ceux présents dans une liste faite par l'école, chaque élève a reçu son thème définitif pour l'élaboration de son travail de bachelor. Dans le cas de ce dossier, le thème est le suivant : **"outhschool.ch, la plateforme Web pour faciliter l'organisation de camps scolaires"**. Ce thème a été placé en choix numéro 1 par l'auteur de ce document lors de l'affectation définitive des thèmes aux élèves, ce qui est une motivation supplémentaire pour mener à bien ce travail.

Cette plateforme est une entreprise fondée via le programme "Business Expérience". Ce dernier offre la possibilité aux étudiants de la HES-SO Valais/Wallis de se lancer dans la conception d'une start-up en les aidant financièrement. Cette entreprise est composée de trois étudiants de la filière Tourisme. Le but de cette plateforme est de faciliter l'organisation de camps scolaires. L'idée est de fournir aux établissements scolaires une plateforme qui permettra de coordonner les différents partenaires et étapes pour l'organisation de camps sportifs. Cette plateforme est en partenariat avec groups.ch qui propose les logements pour groupes dans toute la Suisse. De nombreux autres partenaires potentiels pourront être amenés à contribuer.

1.2 Descriptif du projet

Suite à une discussion une semaine avant le début du travail avec le professeur responsable, le but principal du travail était de réaliser la plateforme Web de A à Z. À ce moment-là, aucune restriction n'était exigée concernant la technologie à utiliser pour la réalisation de cette plateforme. Une réflexion sur la future technologie utilisée avait été faite en préambule du projet.

Le premier jour officiel du travail de bachelor, une séance de démarrage regroupant l'équipe outhschool, le développeur de la plateforme et le professeur responsable s'est tenue afin de fixer les besoins et les fonctionnalités futures de la plateforme. Lors de cette séance, une information peu réjouissante est parvenue : une entreprise externe a été mandatée pour réaliser la version 1.0 de la plateforme !

Cette information a tout changé concernant la réalisation de ce projet. Il ne s'agit plus maintenant de réaliser la plateforme de A à Z, mais de réaliser la version 1.1 qui est la suite logique de la version 1.0. Dès lors, il a fallu changer les objectifs de la plateforme pour la réalisation du travail de bachelor. Après discussion avec l'équipe d'outhschool, des fonctionnalités supplémentaires devront être développées afin d'apporter une plus-value à la plateforme. Les fonctionnalités désirées par l'équipe d'outhschool sont les suivantes :

- ❑ Création d'un gadget Windows 7 pour la gestion des biens
- ❑ Intégration de la base de données de groups.ch (plateforme Web qui regroupe des hébergements pour groupe dans toute la Suisse).
- ❑ Synchronisation des bases de données Jeunesse & Sport des divers cantons

A la suite de cette demande, un cahier des charges a été établi. Seuls les deux premiers points de cette liste ont été intégrés dans ce cahier des charges.

L'enjeu technique de ce travail est de trouver le meilleur moyen de communiquer entre l'application Web et le gadget Windows. Un autre objectif de ce travail est l'intégration des différents acteurs de cette plateforme. Après réflexion, quatre grands acteurs utiliseront la plateforme :

- ❑ Les personnes désirant réserver un camp complet.
- ❑ L'équipe outhschool.ch
- ❑ Les partenaires
- ❑ Les prestataires de services individuels

Les partenaires et les prestataires de services individuels sont deux acteurs qui alimenteront la plateforme avec leurs biens ou leurs services afin qu'une personne désirant réserver un camp puisse choisir les produits de ces deux derniers.

La principale différence entre les partenaires et les prestataires de services individuels est que les partenaires mettent à disposition de la plateforme outhschool, un accès direct à leur base de données. Par la suite, les données seront affichées directement sur la plateforme outhschool.ch en interrogeant la base de données distante du partenaire. Cette solution permet également de pouvoir écrire directement dans la base de données et effectuer une réservation directement sur le système distant.

Quant aux prestataires de services individuels, ce sont des personnes comme tout un chacun qui possèdent un bien et qui désirent le louer dans le cadre d'une réservation sur la plateforme d'outhschool.

De plus, suite à l'annonce de la conception de la version 1.0 de la plateforme, il a également fallu se renseigner sur les différentes technologies utilisées dans la conception de cette version 1.0 de la plateforme. Le langage de programmation utilisé pour la réalisation de la première version de la plateforme est du "**Python**". Tout cela basé sur le framework "**Django 1.1**". Le système de gestion de base de données (SGBD) est basé sur MySQL.

1.3 Motivations personnelles

Possédant déjà de bonnes connaissances dans le développement d'applications Web, je suis très motivé pour mener à terme ce projet et ainsi participer à l'intégration réussie de cette entreprise dans le monde du Web.

Des challenges supplémentaires survenus en début de projet, comme les restrictions technologiques, sont d'autant plus motivants car il n'y a aucune certitude de pouvoir réaliser certaines tâches dans leur intégralité. Tout cela est dû à la méconnaissance du langage.

De plus, les résultats des examens de modules du sixième semestre sont tombés à la mi-juillet. J'ai alors appris que tous les modules ont été passés avec succès. Tout cela signifie que ce travail de bachelor est la dernière chose à valider avant l'obtention du titre délivré à la fin de la formation : "**bachelor en informatique de gestion**". Ceci représente une motivation supplémentaire car ce diplôme me permettra de rentrer dans le monde du travail et ainsi commencer ma carrière d'informaticien.

1.4 Objectifs

Les objectifs principaux de ce travail sont :

- ❑ Gérer de manière autonome un projet informatique
- ❑ Apprendre du langage Python et un nouveau framework de développement Web Django.
- ❑ Récupérer et comprendre un projet existant
- ❑ Prendre les bonnes décisions technologiques
- ❑ Terminer dans les temps avec toutes les fonctionnalités à réalisées.

2 Etat de l'art

Un état de l'art est nécessaire pour la réalisation de ce projet. Il permet de trouver des solutions technologiques et ainsi de définir :

- ❑ **Le langage de programmation**

Malgré le fait que la version 1.0 du site internet a été réalisée avec le framework Django et le langage Python, il a été nécessaire de réfléchir avec quel langage la version 1.1 sera développée parmi tous les langages présents sur le marché.

- ❑ **Le type de Web services**

Cette analyse permettra dans un premier temps de choisir entre les deux types de Web services (REST ou WS-*). Dans un second temps, il permettra de définir quel outil/composant fourni par le langage de développement choisi au point ci-dessus sera le plus adapté à la situation du projet.

- ❑ **Le format des données**

Cette analyse permettra de définir le meilleur langage qui sera utilisé pour l'échange des données entre la base de données de la plateforme Web et le gadget Windows.

2.1 Les langages de programmation Web

2.1.1 Introduction

Parmi tous les sites internet présents sur la toile, on peut clairement en différencier deux types :

- ❑ **Les sites Web statiques**

Ces pages sont généralement créées à l'aide du langage HTML et/ou Javascript. Le contenu de ces pages est dit "statique" car toutes les informations se trouvent directement dans la page elle-même. On parle de langage exécuté coté client. Cela signifie que le client (navigateur Web) interroge le serveur et ce dernier renvoie la ressource demandée au client. Cette ressource est analysée et interprétée par le navigateur Web du client. En terme de sécurité, cette méthode est assez vulnérable.

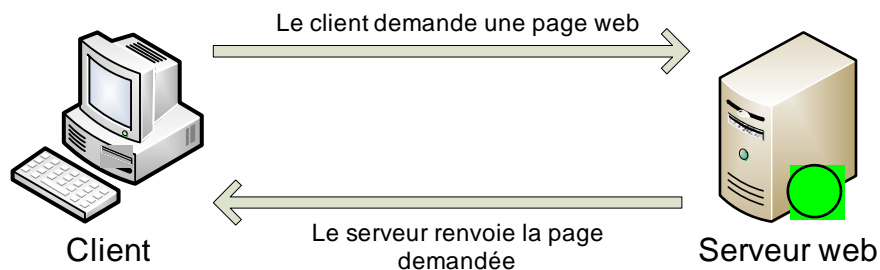


Schéma 1 - Sites Web statiques

- ❑ **Les sites Web dynamiques**

Il existe plusieurs langages de programmation permettant de créer des sites Web dynamiques. Contrairement aux sites statiques, le contenu de sites dynamiques est généré, comme son nom le dit, "dynamiquement". Cela

signifie que les résultats affichés sur la même page peuvent différer selon le choix de l'utilisateur ou les choix du programme lui-même. Ces informations sont généralement stockées dans des systèmes de gestion de bases de données appelés SGBD. Il existe plusieurs SGBD tel que MySQL, SQL Server ou encore Oracle. Mais ce document ne va pas s'étendre en détail sur ces différents SGBD. On parle alors de langage exécuté coté serveur. Cela signifie que c'est le serveur qui analyse la requête du client, ensuite il exécute le code dynamique (PHP par exemple) et lui renvoie seulement le résultat sous forme de fichier contenant seulement du HTML. Le code du langage dynamique n'apparaît donc pas chez le client.

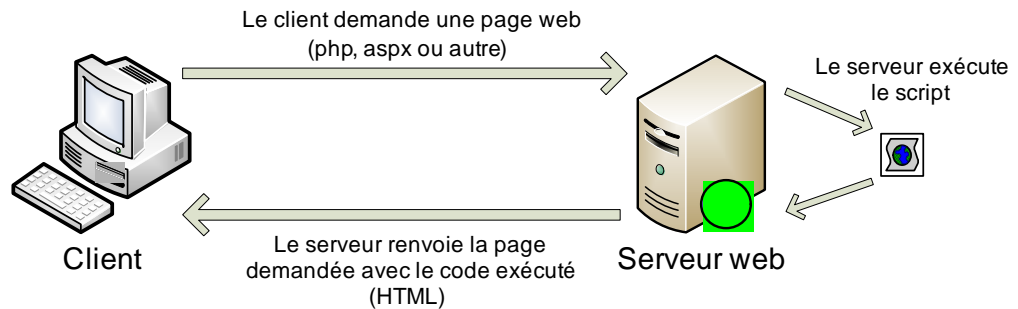


Schéma 2 - Sites Web dynamiques

Ce sont bien sur les langages dynamiques qui sont les plus intéressants car ils offrent beaucoup plus de possibilités dans le traitement et la manipulation de données que le HTML et les autres langages statiques.

2.1.2 Les langages de programmation Web dynamiques

Ci-dessous se trouve une liste plus ou moins exhaustive des différents langages de programmation dynamiques. Pour chacun des langages, un fichier récapitulatif a été faite. On peut remarquer que deux produits sur six sont payants tandis que les autres sont totalement gratuits et certains sont même open source¹.

❑ **PHP** (Hypertext Preprocessor)

Selon les statistiques fournies par TIOBE² au mois d'août 2010, le PHP serait le langage le plus utilisé sur la toile. De plus, il possède des Content Management System, également appelé CMS, très connus et très performants tel que "**Joomla**" ou encore "**Typo3**".

PHP	
Nom complet	Hypertext Preprocessor
Société	The PHP group
Année de parution	1994
Licence	Gratuit
Version actuelle	5.3
Framework	Zend – cakePHP – Symfony
CMS	Joomla – typo3
Site internet	http://www.php.net

¹ Ce dit d'un programme dont le code source est accessible et donc modifiable

² TIOBE est une compagnie spécialisée dans l'évaluation et le suivi de la qualité des logiciels.

❑ PYTHON

Ce langage, tout comme le JSP, est un des langages Web les plus récents. Avec un framework très puissant surnommé "**Django**", Python a fait une excellente entrée dans le monde du développement Web ces dernières années. Il est également intéressant de savoir que Google utilise ce langage pour une partie de ses développements dans différentes applications

PYTHON	
Auteur	Guido van Rossum
Société	Python Software Foundation
Année de parution	1990
Licence	Gratuit
Version actuelle	3.1
Framework	Django – Pylons
CMS	Django-cms
Site internet	http://www.Python.org

❑ PERL

Ce langage est basé sur le langage "**C**" et y reprend ses principales fonctionnalités. Ce langage est particulièrement apprécié dans la manipulation et le traitement des données de fichiers texte grâce à son système de gestion des expressions régulières très performant.

PERL	
Auteur	Larry Wall
Société	The PHP Foundation
Année de parution	1987
Licence	Gratuit
Version actuelle	5.12
Framework	Catalyst – Gantry
Site internet	http://www.perl.org

❑ JSP (Java Server Pages)

Il faut savoir que le JSP n'est pas réellement un langage de programmation mais un framework Web. Comme son nom l'indique, le JSP est basé sur le langage "**Java**". Les pages jsp sont compilées pour se transformer en servlets java.

JAVA – JSP	
Nom complet	Java Server Pages
Société	Sun Microsystem
Licence	Gratuit
Version actuelle	2
Framework	Struts – WebWork
Site internet	http://java.sun.com/products/jsp/

❑ **ASP.NET (Active Server Pages)**

Ce langage est le successeur de l'ASP 3.0 qui est l'ancien langage de programmation Web proposé par Microsoft. En général, le code dynamique utilisé est le "**C#**". Mais les applications Web peuvent également être écrites en Vbscript. Grâce au framework "**.NET**" très ergonomique, il est très aisé de créer des pages Web. De bonnes connaissances du "**C#**" ou du "**Vbscript**" sont conseillées.

ASP.NET	
Nom complet	Active Server Page
Société	Microsoft
Année de parution	2002
Licence	CHF 1'800 (édition standard)
	CHF 15'000 (édition entreprise)
Version actuelle	4
Framework	.NET 4.0 – DotNetNuke
Site internet	http://www.asp.net

❑ **COLDFUSION**

Ce langage est la solution proposée par Adobe pour la réalisation d'application Web. Ce langage est très proche du "**HTML**" contrairement à la plupart des autres langages.

COLDFUSION	
Auteur	Jérémy Allaire
Société	Adobe
Année de parution	1995
Licence	CHF 1'300 (édition standard)
	CHF 7'500 (édition entreprise)
Version actuelle	9
Framework	Struts – WebWork
Site internet	http://www.adobe.com/products/coldfusion

2.1.3 Résultats

On peut dire que tous les langages de programmation Web dynamiques se valent. La concurrence entre eux est très rude, même si on peut dire que le PHP est le plus utilisé de nos jours. Ils possèdent tous les mêmes fonctionnalités de base. La seule différence peut se faire du côté des frameworks qui sont peut-être plus ergonomiques et plus faciles d'utilisation du côté des produits payants. De plus, ces derniers offrent plus de fonctionnalité déjà existante ce qui permet de gagner beaucoup de temps pour la réalisation d'une plateforme Web.

Comme exemple, on peut prendre l'intégration d'un calendrier dans une page Web. En ASP.NET cette fonctionnalité est réalisée à l'aide d'un simple drag-and-drop à l'endroit désiré sur la page. Dans le cas des langages gratuits tel que PHP, il faut

intégrer d'autres libraires / scripts tel que JQuery, ce qui prend beaucoup plus de temps qu'avec un drag-and-drop³ sous Visual Studio.

L'avantages des produits gratuits est certainement du côté de la communauté en ligne qui est très active. Un problème peut être résolu en moins d'une demi-journée. De plus, la plupart d'entre eux sont open source, ce qui permet de pouvoir s'attaquer directement à la structure du langage et modifier certaines fonctionnalités / composants à sa guise.

2.1.4 Décisions prises pour le projet

Etant donné que le serveur sur lequel est hébergé actuellement la version 1.0 de la plateforme (<http://djangohosting.ch/>) et que la future plateforme devra également être hébergée chez ce dernier (pour des raisons de contrat), deux langages de programmation Web étaient à disposition :

- ❑ **PHP**
- ❑ **Python** (framework Django)

Le dilemme qui se posait était le suivant :

- ❑ Recommencer le site à zéro en utilisant du PHP avec la reprise du design existant. Cette solution semblait bien car je possède beaucoup plus de connaissances en PHP. Je développe depuis 3 ans, à côté de mes études, en PHP dans une entreprise qui réalise des sites internet. Mais d'un autre côté, il faut quand même un certain temps pour reprendre le travail effectué sur la version 1.0
- ❑ Soit continuer sur les bases du projet existant sur du Python et le Framework Django. La méconnaissance de ce nouveau langage a installé un certain doute pendant plusieurs jours.

Je me suis posé la question suivante : **"Est-ce que le temps passé à apprendre le langage Python sera suffisant pour refaire toute la plateforme existante en PHP ?"** Après réflexion, je me suis dit qu'apprendre un nouveau langage de programmation serait un plus pour moi et que refaire ce qui existe déjà dans un autre langage juste pour que le travail soit plus facile pour moi était une mauvaise idée.

Au final, la décision d'apprendre ce nouveau langage qu'est le Python a été prise. Ceci augmente la difficulté de ma tâche mais cela représente un énorme challenge qui mérite d'être relevé.

2.2 Les Web services

2.2.1 Introduction

Un Web service est une application internet permettant de communiquer et d'échanger des données entre plusieurs applications. Il est en général composé de plusieurs fonctions. Chacune d'entre elles possède sa propre fonctionnalité de fournir un certain type de données. Les services Web permettent également d'avoir accès en temps réel aux données disponibles sur d'autres systèmes d'informations et d'interagir avec elles sans intervention de l'être humain.

³ Terme anglais pour désigné le glisser-déplacer. Cette pratique permet de prendre un élément et de le déplacer à l'endroit désiré.

Un Web service peut être de type public ou de type privé. Cela signifie que certaines données accessibles depuis l'extérieur peuvent bien sûr être protégées par un système d'authentification.

Deux grands types de Web service existent actuellement dans le monde du Web :

❑ **Type REST (Representational state transfer)**

Ce type de Web service propose toutes ses fonctionnalités en tant que ressource (appelée URI) accessible via le protocole HTTP. Il est donc basé sur l'architecture de l'internet.

Le protocole http offre quatre principales méthodes pour interagir avec ces ressources disponibles sur un Web service de type REST

- GET permet de lire une / des ressource(s)
- POST permet d'ajouter une ressource
- PUT permet de modifier une ressource
- DELETE permet de supprimer une ressource

❑ **Type WS**

Ce type de Web service propose exactement les mêmes fonctionnalités mais cette fois non pas sous forme de ressource mais sous forme d'exécutables à distance. Ce système utilise les standards SOAP (Simple Object Access Protocol) employés pour l'échange des données et WSDL (Web Service Description Language) pour tout ce qui concerne la description du service en lui-même. On entend par là, la description des fonctionnalités, celle des fonctions disponibles, celle des protocoles utilisés et également celle du type de données. Il est basé sur une architecture de type SOA (System Oriented Architecture).

En résumé, les deux types de Web service proposent deux architectures totalement différentes. L'un est basé sur une architecture Web et l'autre sur une architecture SOA. Étant donné que le Web est plus universel, moins de problèmes sont rencontrés avec des Web services de type REST dans la capacité de s'intégrer et d'opérer avec plusieurs systèmes d'informations hétérogènes.

2.2.2 Les outils existant pour le Framework Django

Il existe deux composants Django permettant la création de Web service. Il s'agit de **Django-piston** et de **Django-rest-interface**. Ces deux composants proposent la création de Web service de type REST. Le tableau ci-dessous démontre les différentes fonctionnalités que propose chacun des composants.

Fonctionnalité	Piston	Rest-Interface
Intégration dans des projets django	●	●
Ressources basées sur des objets	●	●
Ressources personnalisables	●	●
Création de Web service publics	●	●
Création de Web service privés		
Authentification HTML	●	●
Authentification avec OAuth ⁴	●	●
Langage de manipulation de données		
XML	●	●
JSON	●	●
YAML	●	●
HATEOAS	●	●
Génération de la documentation	●	●
Difficultés de mise en place	2/5	4/5
Restriction sur les méthodes HTTP	●	●

2.2.3 Les langages d'échange de données

Il existe plusieurs sortes de langage permettant l'échange de données entre les systèmes d'informations. Le plus connu d'entre eux est le XML. Mais d'autres langages tels que le JSON ou le YAML sont également présents sur le marché. Tous ces langages sont utilisés pour manipuler tous types de données uni codes.

❑ XML (Extensible Markup Language)

Les principaux avantages :

- Il possède une meilleure structure que les autres langages
- Le code est facile à comprendre
- Tous les jeux de caractères sont supportés (uft-8 ou autre)
- Il est reconnu et utilisé par tous les langages de programmation

⁴ Protocole libre permettant une authentification à une API (<http://fr.wikipedia.org/wiki/OAuth>)

Le principal inconvénient :

- Un fichier XML peut rapidement devenir très lourd par rapport à d'autres formats binaires ou des formats sous forme de tableaux comme le JSON

```
<?xml version="1.0" encoding="UTF-8"?>
<developpeurs>
  <developpeur nom="Pache" prenom="Barnabé">
    <projets>
      <projet>Web 3.0</projet>
      <projet>Outschool.ch</projet>
    </projets>
  </developpeur>
</developpeurs>
```

Extrait de code 1 - Exemple XML

□ JSON (Javascript Object Notation)

Les principaux avantages

- Il est facile à utiliser en Javascript car il possède la structure d'objets Javascript
- Il est reconnu et utilisé par de plus en plus de langages de programmation

Le principal inconvénient

- Pour travailler avec un document, l'utilisateur est obligé de connaître la structure de l'objet avec lequel il travaille

```
{
  developpeurs: {
    developpeur: {
      nom: "Pache"
      prenom: "Barnabé"
      projets: [
        {
          projet: "Web 3.0"
        },
        {
          projet: "Outschool.ch"
        }
      ]
    }
  }
}
```

Extrait de code 2 - Exemple JSON

□ YAML (Ain't Markup Language)

Les principaux avantages

- Il est très léger
- Il est très lisible, car aucune balise n'est utilisée
- Il est possible d'y intégrer de la syntaxe JSON

Les principaux inconvénients

- Sa structure est basée sur l'indentation donc très complexe
- Pas beaucoup de langages de programmation sont munis de parseur YAML.

```
developpeurs:
  - nom      : "Pache"
    prenom   : "Barnabé"
    projets:
      - projet : "Web 3.0"
      - projet : "Outschool.ch"
```

Extrait de code 3 - Exemple YAML

2.2.4 Résultats

Au niveau des composants django pour la création de Web services, on peut en déduire, d'après le tableau ci-dessus, que **django-piston** est beaucoup **plus complet** et propose beaucoup plus de fonctionnalités que son concurrent direct **django-rest-interface**. Il propose également plus de langages permettant le transfert de données. Mais les fonctionnalités de base d'un Web service sont présentes chez les deux composants.

Au niveau des langages d'échanges de données, les principaux acteurs de ce domaine sont le XML et le JSON. Au vu des avantages cités ci-dessus, le XML offre beaucoup plus de solutions que ses concurrents. L'utilisation de l'un ou l'autre de ces langages dépend souvent des technologies utilisées dans un projet.

2.2.5 Décisions prises pour le projet

Après plusieurs tests effectués dans le cadre de ce travail, il est totalement juste d'affirmer que django-piston est beaucoup plus difficile à mettre en place et à utiliser. La génération de la documentation n'a jamais fonctionné dans le cadre de ce projet. Cette dernière fonction était un des points forts de django-piston. N'arrivant pas à la faire fonctionner, j'ai donc décidé d'utiliser **django-rest-interface**. Les fonctionnalités de ce composant suffisent entièrement à la réalisation de ce projet.

Etant donné que le composant utilisé, soit django-rest-interface, ne propose que le XML et le JSON comme formats d'échanges de données, des tests ont été réalisés avec ces deux langages. Le test de lecture du XML en Javascript a été très satisfaisant et très rapide. Cela est certainement dû à de bonnes connaissances de ce langage. Comme je ne connaissais pas du tout le JSON, l'apprentissage a été plus difficile et moins convaincant avec quelques problèmes non résolus. Malgré plusieurs tutoriaux suivis sur le net, je n'ai même pas réussi à lire un objet JSON avec JavaScript et la fonction "eval()" qui est prévue à cet effet ! Pour éviter de passer encore des heures sur ce problème, mon choix c'est rapidement porté sur le **XML**.

3 Le projet outschool.ch

3.1 Description détaillée du projet

Pour rappel, la plateforme outschool.ch doit permettre de faciliter l'organisation de camps scolaires aux professeurs d'école de la Suisse entière. C'est, en résumé, un site de réservation de différents biens ou services proposés par des partenaires ou des prestataires de services individuels. Ces biens et services sont répartis dans différentes catégories telles que : Hébergement, Moniteur, Transport et Matériel.

- ❑ **Un partenaire** est une "entreprise Web" qui propose **plusieurs biens** d'une catégorie. Par exemple groups.ch, qui propose des centaines d'hébergements situés dans toute la Suisse.
- ❑ **Un prestataire de services individuels** est un particulier qui désire mettre à disposition des personnes organisant des camps **un bien** qui lui appartient. Par exemple, une personne possédant une grande maison qui permettrait de loger une vingtaine de personnes pourrait être utilisée par un groupe.

Il est très important pour la suite de la lecture de bien comprendre la différence entre ces deux types d'acteurs.

Concernant l'intégration des partenaires, il a fallu réfléchir sur une solution qui permette l'intégration des données de la base de données des partenaires directement sur la plateforme outschool.ch. Car groups.ch avait demandé à ne pas être impliqué dans cette intégration au niveau de ses employés et en particulier des informaticiens.

Avec ces différentes restrictions, la meilleure solution qui est ressortie est la suivante : le partenaire devra mettre à disposition un accès à sa base de données en production. Par la suite, un outil de mapping sera utilisé pour mettre en relation la base de données du partenaire avec celle d'outschool et finalement pouvoir intégrer facilement les données du partenaire sur la plateforme outschool.ch. Avec ce système, groups.ch n'a besoin que de 3 minutes pour s'inscrire sur la plateforme et renseigner les informations nécessaires à la connexion à distance sur sa base de données (URL, type de base de données, utilisateur et mot de passe).

Concernant l'intégration des prestataires de services individuels, un outil de gestion est donc nécessaire afin que ces derniers puissent ajouter, modifier ou supprimer un bien leur appartenant. Une autre fonctionnalité de cet outil est d'avoir un aperçu visuel des occupations d'un bien sur le calendrier. Sur proposition de l'équipe outschool, un gadget Windows sera utilisé comme outils de gestion. Cette idée peut paraître un peu bizarre mais finalement cet outil est intéressant car l'utilisateur n'a pas besoin de se connecter via un navigateur Web sur une plateforme, la gestion se fait directement depuis son bureau Windows. De plus le gadget permet d'obtenir des informations en temps réel ce qui peut être utile pour afficher les nouvelles réservations sur son bien par exemple.

La grande force de la plateforme est qu'elle est **générique**, cela signifie qu'une fois le développement de cette plateforme terminé, aucune intervention (ou presque) de la part d'un développeur Web n'est nécessaire pour ajouter l'un ou l'autre des éléments présents sur la plateforme.

3.2 Fonctionnalités déjà existantes

Une version 1.0 du site a été développée et mise en production les 2 juin 2010. Le code source du site a donc été remis peu après cette date (3 semaines après le début du travail de bachelor). La plateforme contenait à ce moment-là les fonctionnalités suivantes :

- ❑ Un module de news
- ❑ Un formulaire de contact
- ❑ Réservation d'un package "all-inclusive"
- ❑ Réservation d'un package personnalisé

Après l'analyse du code reçu au début du mois de juin 2010, le module de news, le formulaire de contact et la mise en place du CMS (django-cms) étant fonctionnel, ces éléments ne vont pas être modifiés et gardés tels quels pour ce projet. Au niveau de la réservation de package "all-inclusive", peu de changements vont être réalisés car toutes les informations sont déjà stockées dans la base de données. Par contre, au niveau de la réservation personnalisée, tout est à refaire. A ce moment-là le système ne sauve rien dans la base de données, il ne fait qu'envoyer un mail avec les diverses informations d'un formulaire "en dur" concernant la réservation souhaitée par l'utilisateur. Le traitement se fait ensuite manuellement pas l'équipe d'outhschool.

The screenshot shows a web form titled "INFORMATIONS" in a dark blue header. Below the header, the reservation details are listed: "De : 16/08/2010", "A : 19/08/2010", "Nombre total de personnes : 13", and a bulleted list: "• Nombre d'élève : 12" and "• Nombre d'accompagnants : 1". Below this, the text "Veuillez sélectionner les prestations désirées:" is followed by five rows, each with a label and a checkbox: "Hébergement : ☒", "Transport : ☐, "Cuisinier : ☐, "Moniteur(s) : ☐, and "Matériel : ☐". At the bottom of the form is a blue button labeled "Valider les prestations".

Ecran 1 - Formulaire de réservation (version 1.0)

3.3 Architecture

Ci-dessous se trouve une représentation visuelle qui liste les différents éléments mis en place pour la réalisation du projet outhschool. Grâce à l'outil de mapping objet relationnel, la couche de données contient les classes métiers. Pour le Web service, des collections faisant appel aux classes métiers sont utilisées pour manipuler les enregistrements provenant de la base de données. Concernant l'accès aux bases de données externes, les connexions sont ouvertes au moment où la plateforme fait appel aux données, c'est-à-dire dans le formulaire de réservation.

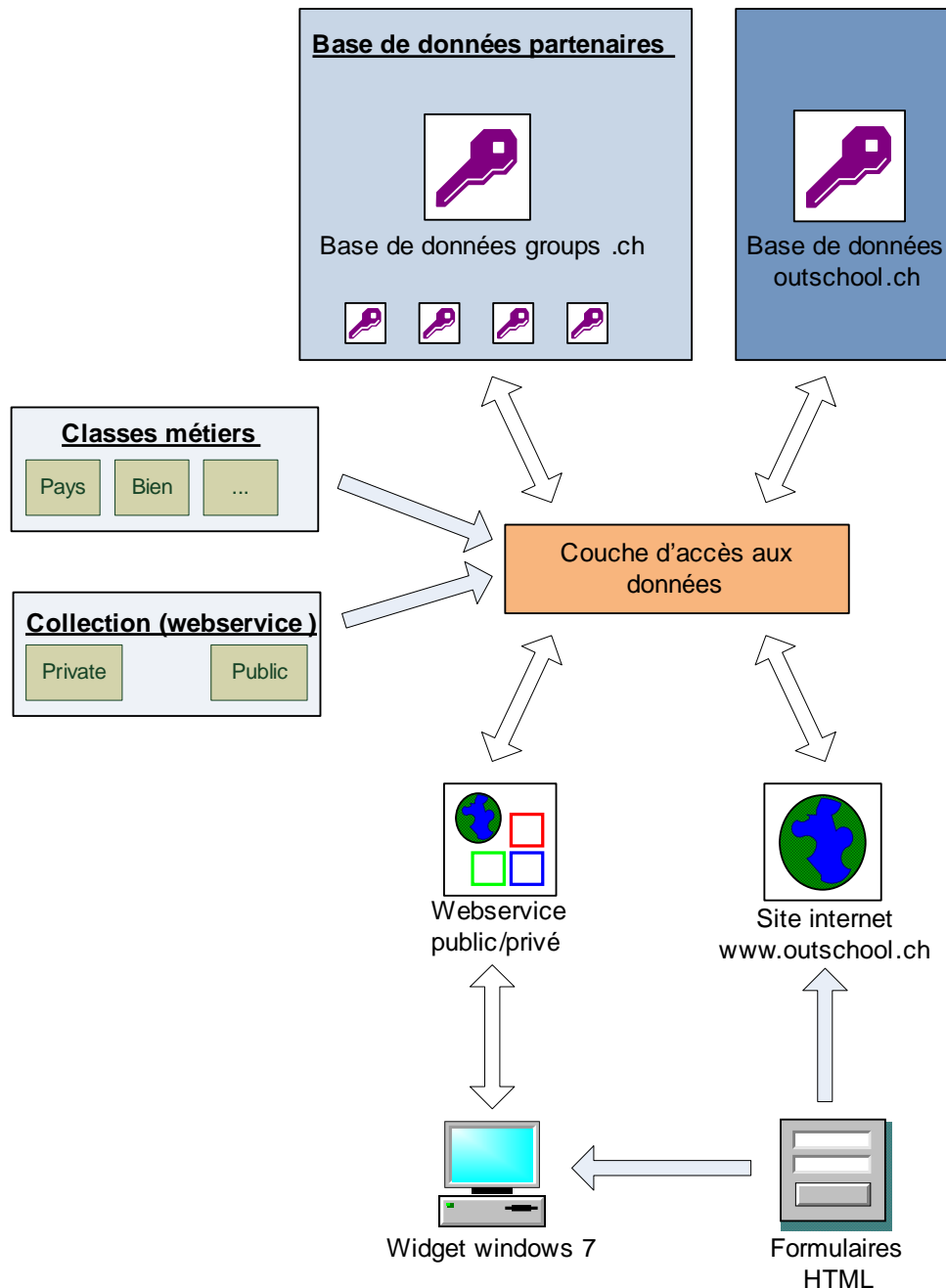


Schéma 3 - Architecture de l'application

3.4 Diagramme de classes

Comme cité dans la description ci-dessus, l'application se veut d'être **générique**. Ce diagramme de classe permet d'être partiellement générique surtout au niveau des prestataires de service et de leurs biens. On remarque que la classe "Bien" ne contient pas de champs car tous les champs sont générés automatiquement avec la classe "Champs". Vous pouvez voir sur le schéma ci-dessous toutes les classes, les variables et les relations entre les classes qui sont utilisées pour la plateforme.

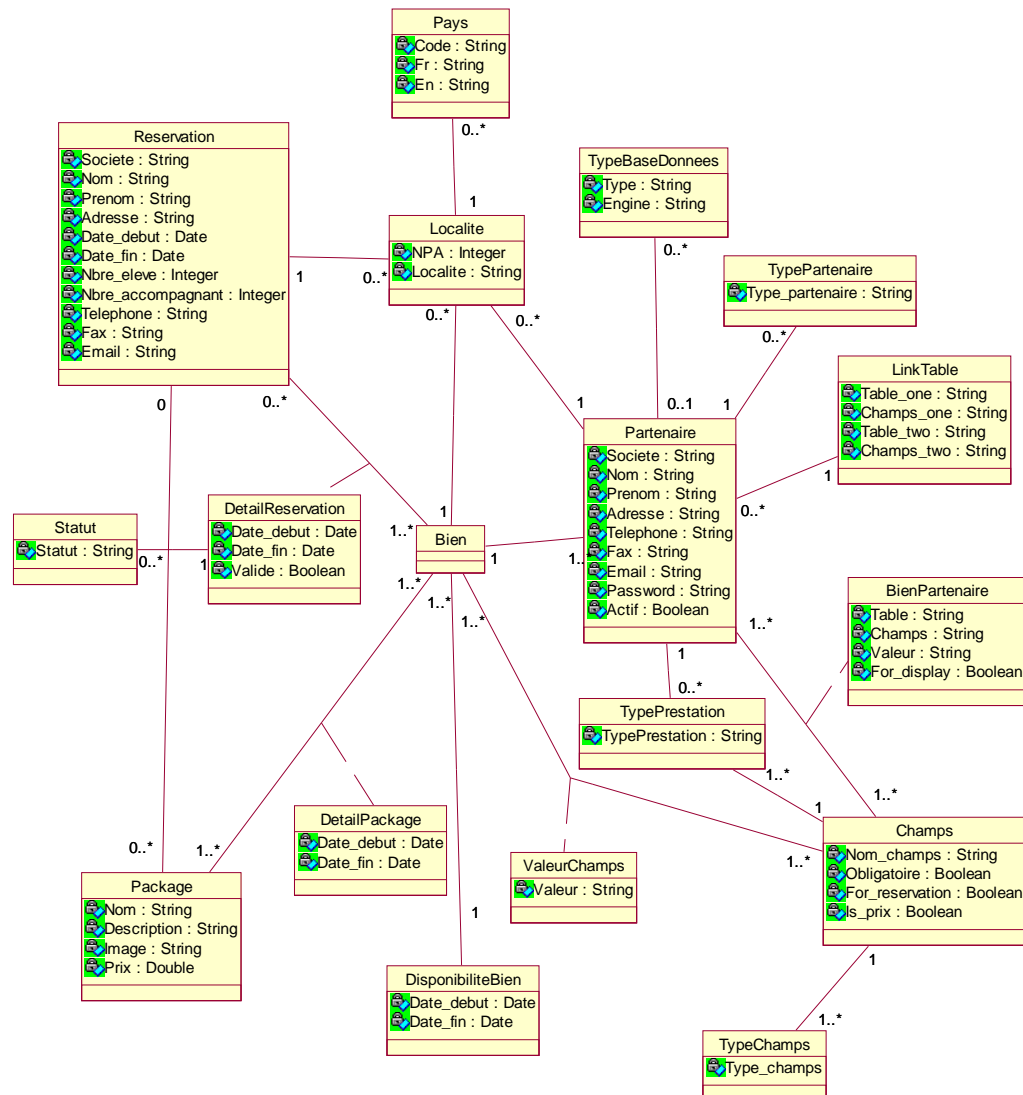


Schéma 4 - Diagramme de classes

4 Développement de la plateforme Web

4.1 Fonctionnalités

4.1.1 Recherche multicritère de packages

Cette fonctionnalité permettra à l'utilisateur final de filtrer les résultats des packages "all-inclusive". Cette fonctionnalité a été demandée par l'équipe outschool car elle n'a pas été réalisée pour la version 1 de la plateforme. C'est d'ailleurs la seule modification qui a eu lieu sur la partie des packages "all-inclusive". Il a fallu bien analyser et comprendre le code existant pour connaître son fonctionnement et ajouter le code nécessaire aux bons endroits afin d'obtenir le résultat ci-dessous. A noter qu'aucun champ n'est obligatoire et que si l'utilisateur envoie le formulaire vide, le système retournera tous les packages existants.

Le filtre peut être réalisé sur les critères suivants :

- ☐ Le nombre de personnes
- ☐ Le prix en francs suisses par personne
- ☐ Savoir si les différents services sont inclus dans le package
 - ☐ Transport
 - ☐ Cuisine
 - ☐ Moniteur
 - ☐ Equipement



Ecran 2 - Formulaire de recherche multicritère

4.1.2 Formulaire d'inscription pour les partenaires et prestataires

Afin que les différents acteurs du contenu de la plateforme, soit les partenaires et les prestataires de services individuels, puissent y enregistrer leurs biens et services, ils doivent tout d'abord s'inscrire via le formulaire prévu à cet effet dans la page partenaire du site.

Ce formulaire contient toutes les informations relatives à un partenaire ou prestataire de services afin de pouvoir contacter les personnes en cas de problème.

Le champ "Type partenaire" est rempli dynamiquement avec la table du même nom dans la base de données. Elle contient également une info-bulle. Quand l'utilisateur clique sur cette dernière, une fenêtre apparaît et explique la différence entre un partenaire et un prestataire de services.

Les champs "Type prestation" et "Localité" sont également remplis dynamiquement depuis la base de données.

COORDONNÉES PERSONNELLES

Type partenaire * Ce champs est obligatoire

Type prestation *

Societe

Nom *

Prenom *

Adresse *

Localite * Ce champs est obligatoire

Telephone * Ce champs est obligatoire

Fax

Email * Le format est incorrect

Les champs marqués d'un * sont obligatoires !

Ecran 3 - Formulaire d'inscription

Lorsque l'utilisateur clique sur le bouton "S'inscrire", un contrôle automatique est effectué sur tous les champs obligatoires. Si ces derniers ne sont pas remplis, un message rouge apparaît pour signaler à l'utilisateur que le formulaire n'a pas été correctement rempli.

Un contrôle est également fait sur les champs "Email". Si l'adresse email n'a pas un format correct, une erreur s'affiche indiquant que le format n'est pas correct.

Si tous les champs ont été correctement remplis, un nouvel enregistrement est inséré dans la base de données. Pour des raisons de sécurité, un identifiant encodé est généré à la création d'un compte. Cet identifiant est composé de l'adresse email rentrée dans le formulaire, du mot de passe de l'utilisateur (chaîne de six caractères générée aléatoirement lors de l'inscription) ainsi que la date et l'heure de l'inscription. Ces trois paramètres permettent d'obtenir un identifiant unique pour chaque enregistrement. Pour finir, cette chaîne de caractères est cryptée à l'aide de l'algorithme MD5. Le mot de passe de six caractères est lui également crypté à l'aide de cet algorithme.

A ce moment-là, l'utilisateur ne peut pas encore être utilisé car il doit être validé par l'utilisateur. Pour ce faire, un courriel est envoyé à l'adresse fournie dans le formulaire (ce qui permet de contrôler si l'utilisateur est bien propriétaire de l'adresse email).

A l'intérieur de ce courriel, se trouvent deux liens. Le premier permet d'activer le compte et le second permet à l'utilisateur de changer son mot de passe qui a été, auparavant, généré automatiquement. Il faut noter que la partie du mot de passe est visible seulement sur le courriel envoyé aux prestataires de services individuels. Les partenaires n'ont pas besoin de mot de passe étant donné qu'il donne l'accès à leur base de données.

On peut remarquer dans l'exemple de courriel ci-dessous, la présence de l'identifiant encodé à la fin de chaque adresse internet également appelé URL.



Ecran 4 - Courriel de confirmation

Quand l'utilisateur clique sur le lien de confirmation (1^{er} lien), il est redirigé vers la plateforme outschool.ch. Un simple message de confirmation s'affiche. S'il clique sur le 2^{ème} lien, il sera redirigé vers le formulaire permettant de modifier son mot de passe. Ce formulaire demande une double validation du mot de passe.

A screenshot of a web form titled 'nouveau mot de passe' in a blue header. The form has three input fields: 'Adresse email' with the value 'barnabe_pache@hotmail.com', 'Nouveau mot de passe *' with masked dots, and 'Retaper le mot de passe *' which is empty. To the right of the password fields, there are two red error messages: 'Les mots de passe ne sont pas identiques'. At the bottom, there is a grey 'Enregistrer' button.

Ecran 5 - Formulaire de modification de mot de passe

Attention : Si le type de partenaire choisi dans la liste déroulante du formulaire d'inscription est "Partenaire" cela signifie qu'il désire mettre à disposition un accès sur sa base de données. A ce moment-là, il devra remplir un autre formulaire concernant toutes les informations pour la connexion à distance sur sa base de données.

Le champ "Type de base de données" est rempli dynamiquement depuis la base de données et contient également la chaîne de caractères permettant de se connecter à une base de données externes avec le framework Django.

Informations base de données

Type bdd * MySQL

Site web

BDD uri * http://mysql.outschool.ch

BDD name * outschool

BDD login * Ce champs est obligatoire

BDD password * Ce champs est obligatoire

Enregistrer

Les champs marqués d'un * sont obligatoires !

Ecran 6 - Formulaire d'informations sur la base de données

Il a fallu réfléchir à des solutions alternatives au cas où le partenaire ne désire pas délivrer ces informations. Peu de solutions ont été trouvées car il n'existe pas énormément de solutions pour obtenir les informations en temps réel d'un système d'informations externes (accès direct à la base de données ou accès à un Web service). Les solutions citées ci-dessous ne seront pas développées dans le cadre de ce travail, car le temps mis à disposition ne le permet pas!

La première solution étudiée serait de demander aux partenaires les accès à un Web service qui délivreraient également les informations sur les biens et pourraient être ensuite affichés sur la plateforme outschool.ch.

La seconde solution serait de persuader le partenaire de donner ses informations. L'utilisateur aurait la possibilité de faire une demande de démonstration à l'équipe d'outschool. A ce moment-là, l'équipe outschool devrait demander une copie de la base de données au partenaire afin de l'installer sur un serveur de développement, pour ensuite se rendre chez le partenaire afin de lui faire une démonstration et ainsi espérer qu'il soit convaincu de mettre à disposition les accès à sa base de données.

4.1.3 Formulaire de réservation

Le formulaire de réservation qui sera utilisé par des personnes désirant réserver des camps personnalisés est décomposé en quatre grandes parties. A chaque étape du processus, les données sont à chaque fois enregistrées dans une variable de session, ce qui permet de les modifier en tout temps. Cela signifie que l'utilisateur pourra sans aucun problème revenir sur une étape du processus et y retrouver les informations précédemment saisies. Cette variable de session sera détruite à la fin du processus de réservation.

Pour cette partie, il faut noter que pour l'intégration des biens de la base de données des partenaires, tout n'a pas pu être réalisé car la base de données de groupes nous a été remise dix jours avant la fin de ce travail.

❑ Les informations générales sur le camp

Ces informations seront très importantes car elles seront utilisées dans l'étape suivante pour afficher les services par rapport à ces critères. Pour la sélection des dates, le "**datepicker**" de la librairie "**Jquery**" a été utilisé. Un contrôle est effectué sur tous les champs afin de vérifier s'ils sont bien remplis.

Informations générales

Date de début * 2010-08-16

Date de fin * 2010-08-20

Nombre d'élèves * 12

Nombre d'accompagnants * Ce champ est obligatoire.

Destination désirée * Crans-montana

Les champs marqués d'un * sont obligatoires!

Etape suivante

Ecran 7 - Formulaire de réservation → Saisie des informations principales

❑ Les services désirés lors de ce camp

A cette étape, les informations saisies dans l'étape 1 seront utilisées pour filtrer les différents services à disposition. Ce filtre est effectué sur tous les champs saisis à l'étape précédente, c'est-à-dire les dates, la capacité et la destination.

Au sommet de l'écran, un résumé de l'étape 1 est affiché. L'utilisateur peut en tout temps cliquer sur "Modifier" pour modifier les paramètres de base. Attention, cette opération supprimera tous les services choisis car ils dépendent évidemment des critères de l'étape une. Par exemple, si l'utilisateur change la destination, les propositions de services ne seront plus les mêmes.

Sur cette page et pour des raisons d'organisation et d'ergonomie, tous les types de services sont affichés chacun dans un onglet. C'est à cet endroit que sont affichés les biens et services des prestataires de services individuels et ceux des partenaires. On remarque sur l'écran ci-dessous qu'un message s'affiche lorsque le bien provient d'un partenaire, c'est-à-dire d'une base de données externe. On remarque également que pour le bien provenant d'un partenaire, tous les champs s'affichent. Cela signifie que tous les champs n'ont pas pu être mappés avec les champs d'un bien de type "Hébergement".

Pour l'hébergement, l'utilisateur ne peut sélectionner qu'un bien tandis que pour les autres services (moniteur, matériel et transport) l'utilisateur peut sélectionner plusieurs choix à l'aide de check box.

Informations générales [Modifier](#)

Du : 2010-08-16
 Au : 2010-08-20
 Destination : Crans-montana
 Nombre total de personnes : 14

- Nombre d'élève : 12
- Nombre d'accompagnants : 2

Hébergement | **Moniteur** | **Matériel** | **Transport**

Cet élément provient d'un partenaire

<input type="radio"/>		Nom Ma Maison Blanche (MMB) Capacité 50 Description Joli maison Nombre de salles 20
<input type="radio"/>		Nom Cabane du ski club Capacité 35 Description Située au dessus de la station de Crans-Montana. 10 min du centre Prix/nuit/personne (chf) 25 Site internet - Nombre de salles -

[Etape suivante](#)

Ecran 8 - Formulaire de réservation → Sélection des services

□ **Les coordonnées de contact**

Ce formulaire permet d'obtenir les coordonnées de contact de la personne faisant une réservation. C'est un formulaire simple avec un contrôle des champs obligatoires. Le champ "Localité" est rempli dynamiquement depuis la base de données.

COORDONNÉES PERSONNELLES

Ecole/institution: HES-SO Valais / Wallis

Nom *: Doche

Prenom *: Florian

Adresse *: Ce champs est obligatoire

Locality *: Sierre Ce champs est obligatoire

Telephone *: Ce champs est obligatoire

Fax:

Email *: Ce champs est obligatoire

Ecran 9 - Formulaire de réservation → Coordonnées de contact

□ Confirmation

Avant de confirmer, l'utilisateur a la possibilité de voir un aperçu global de toutes les prestations sélectionnées. Les coordonnées de contact de chaque prestataire sont également affichées sur ce résumé

INFORMATIONS GÉNÉRALES

Du : 2010-08-16
Au : 2010-08-20
Destination : Crans-montana
Nombre total de personnes : 14

- Nombre d'élève : 12
- Nombre d'accompagnants : 2

Hébergement : Bien object Cabane du ski club : Cordonier Jessica - 07974622383
Moniteur :

- Cours de Golf : Pache Barnabé - 0797641532

Matériel : Aucun matériel
Transport : Aucun transport
Coordonnées de contact

- Ecole/institution : HES-SO Valais / Wallis
- Nom : Doche
- Prénom : Florian
- Adresse : Bellevue 12
- Localité : Sierre
- Téléphone : 0274543423
- Fax :
- Email : barnabe.pache@gmail.com

Ecran 10 - Formulaire de réservation → Confirmation

Une fois que l'utilisateur a cliqué sur "Terminer et confirmer", la réservation est enregistrée dans la base de données. Tous les biens sélectionnés sont également marqués dans la base de données comme réservés pour la date

du camp. Le statut "non-lu" est mis par default à chaque réservation. Ce statut évolue selon les choix des prestataires de service. Cette partie sera développée dans la partie du gadget Windows.

Egalement à ce moment-là, la variable de session est détruite et les informations ne sont donc pas récupérables. Tous les formulaires sont à nouveau vierges. De plus, lors de la confirmation, un email est envoyé à la personne ayant fait la réservation. Dans ce mail se trouve un lien qui permet de voir l'avancement des différentes demandes de services ou de biens.

Cabane du ski club	Pas traité
Personne de contact : Jessica Cordonier - 07974622383	
Cours de Golf	Accepté
Personne de contact : Barnabé Pache - 0797641532	

Ecran 11 - Suivi d'une réservation

4.1.4 Zone d'administration

Le framework Django fournit un outil qui permet de générer une zone d'administration par rapport aux classes métiers que le développeur a créées. Il suffit de spécifier le nom de la classe de l'objet que l'on désire modifier via l'interface d'administration Web pour voir apparaître dans la zone d'administration les différentes actions possibles sur tous les objets de cette classe. Les différentes actions possibles sont les suivantes : ajouter, modifier et supprimer un objet.

Un outil de mapping avec les bases de données externes des partenaires devait être réalisé pour ce travail. Suite à un retard de la réception de la base de données de groups.ch et surtout à un manque de temps durant ce travail, cet outil n'a pas pu être réalisé comme voulu. Mais la structure de la base de données est déjà prévue pour réaliser cet outil. D'ailleurs, les tables prévues à cet effet ont été remplies manuellement afin de pouvoir quand même afficher les données depuis une base de données distantes dans le formulaire de réservation.

4.1.5 Le Web service

La mise en place d'un Web service a été nécessaire afin de pouvoir permettre la communication entre la base de données de la plateforme outhschool.ch et l'outil de gestion des biens pour les prestataires de services individuels. Il s'agit du gadget Windows 7.

Ce Web service pourra également être utilisé pour d'autres outils de gestion. L'équipe d'outhschool m'a déjà demandé de faire un gadget pour Mac étant donné qu'un certain nombre de professeurs utilise des Mac et n'ont donc pas accès à Windows. A ce moment-là, toutes les fonctionnalités existent déjà sur le Web service et il sera donc plus aisé pour la réalisation de ce gadget pour Mac.

Ce service a été réalisé à l'aide du composant "django-rest-interface" conçu pour fonctionner avec le framework Django. Il permet de créer assez aisément des Web services de type REST. Il a été séparé en 2 parties :

- ❑ **Un Web service public** qui sera utilisé pour effectuer l'authentification depuis le gadget Windows 7 sur la base de données. Cela signifie que l'utilisateur, avant de pouvoir utiliser l'outil de gestion de contenu, va devoir s'authentifier avec son adresse email et le mot de passe qui a été généré lors du processus d'inscription. Si l'authentification réussit, le système va pouvoir afficher toutes les informations nécessaires en utilisant la partie privée du Web service. Pour des raisons de sécurité, seule la méthode GET, qui permet la lecture de ressources, est autorisée à l'appel de cette fonction.
- ❑ **Un Web service privé** qui requiert une authentification HTTP a été créé afin de sécuriser toutes les données sensibles. Sur certaines fonctionnalités, il n'y a pas seulement la méthode GET qui est utilisée mais également les méthodes POST, PUT et DELETE. Ces dernières permettent d'ajouter, modifier et supprimer des ressources c'est pourquoi le Web service a été sécurisé. Par défaut, le composant vérifie l'authentification sur une table appelée "django_users" créée par Django. C'est pourquoi le composant a subi des modifications pour que l'authentification se fasse sur la table "Partenaire" de la plateforme outschool.

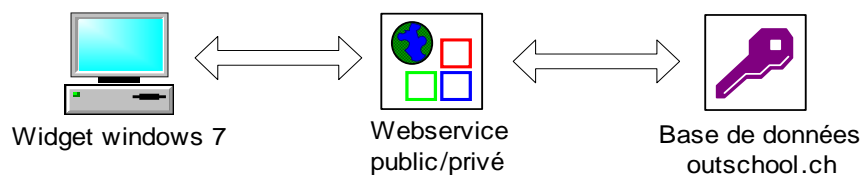


Schéma 5 - Fonctionnement du Web service

4.2 Technologies utilisées

4.2.1 Framework Django

Le framework Web **Django** a été développé il y a 4 ans. Il a été écrit entièrement en Python. Très puissant, il a été utilisé par des organisations comme la "NASA" et le "Washington Times" ainsi que le "Google App Engine⁵" qui a été en partie développé avec ce framework. La version actuelle de ce framework est la 1.2 sortie en mai 2010. Son slogan est le suivant : "Le framework Web pour les perfectionnistes sous pression". Ce qui laisse entrevoir un travail rapide et bien fait.

Django est un framework assez simple d'utilisation grâce à la bonne documentation que propose différents sites. Il est quand même nécessaire d'avoir d'excellentes connaissances de programmation Web et de programmation orientée objet (POO). Django est basé sur le système "Modèle-Vue-Contrôleur" (MVC) ce qui permet d'avoir chaque partie de l'application séparée l'une de l'autre. Il fournit également d'autres fonctionnalités très utiles comme par exemple:

⁵ Plateforme de conception et d'hébergement d'applications web basée sur les serveurs de Google (http://fr.wikipedia.org/wiki/Google_App_Engine)

- ❑ **Un mapping objet-relationnel** est utilisé par ce framework. Cette technique permet de travailler directement avec les enregistrements de la base de données tout en ayant l'impression de travailler avec les objets de l'application.
- ❑ **Une API d'accès aux données** est intégrée au framework. Plus besoin d'avoir de connaissances en SQL⁶. Il est également possible de faire ses propres requêtes SQL si le développeur ne veut pas utiliser cet API. Avec cet outil, 98% des requêtes peuvent être effectuées.

API Django	SQL
MaClasse.objects.all()	SELECT * FROM MaClasse
MaClasse.objects.filter(id=1)	SELECT * FROM MaClasse Where id=1

- ❑ **Une interface d'administration automatisée** peut être activée en 1 minute et ensuite être accessible par l'adresse "/admin/". Tous les formulaires nécessaires à l'ajout, la modification et la suppression d'enregistrement sont générés automatiquement.
- ❑ **Le design d'URL** est assez particulier car les URL ne correspondent pas à des ressources physiques sur le serveur. Un fichier nommé "URLs.py" gère toutes les URL d'une application Web. Chaque URL présente dans ce fichier font directement appel au contrôleur qui lui-même fait appel à la vue.
- ❑ **Un mini serveur Web** qui permet de développer ces applications en local sans avoir besoin d'utiliser Apache ou un autre serveur Web.

En annexe se trouve un document qui explique toutes les étapes à suivre pour mettre en place l'environnement de travail et ainsi bien commencer avec Django.

4.2.2 Environnement de développement

L'environnement de développement (IDE) utilisé est **Eclipse 5.3 (Galileo)** avec le plugin **PyDev** qui permet le développement d'application Python mais également de plateforme Web de type Django.

Fiche technique	
Année de parution	2001
Développement	Eclipse Foundation
Version actuelle	3.6 Helios (23.06.2010)
Licence	EPL
Site internet	www.eclipse.org

⁶ Langage de programmation permettant d'interroger des bases de données

4.2.3 SGBD

Le système de gestion de base de données (SGBD) utilisé pour stocker les données de la plateforme outschool.ch est de type "**MySQL 5.0**". Le serveur de production ne proposant seulement deux types de SGBD soit MySQL et PostgreSQL, mon choix s'est vite porté sur MySQL car je le connais beaucoup mieux. Par ailleurs, ce dernier a été utilisé dans la version 1.0 de la plateforme.

Fiche technique	
Année de parution	1994
Développement	Sun Microsystems
Version actuelle	5.1 (17.06.2010)
Licence	GNU
Site internet	www.mysql.com

4.2.4 Serveur Web

Le serveur Web utilisé pour le développement de l'application est le petit serveur Web intégré au framework Django. Certes un peu lent, il remplit totalement et avec satisfaction les besoins d'un serveur de développement.

Pour le serveur de production, la plateforme sera hébergée chez www.djangohosting.ch qui propose des hébergements pour les sites internet développés à l'aide de Django. Le serveur Web sera alors de type "**Apache**". Cet hébergeur support également le PHP. Un contrat d'une année a été signé entre l'équipe outschool et ce dernier.

5 Développement du gadget Windows 7

5.1 Qu'est-ce qu'un gadget ?

Un gadget Windows (également appelé widget) est un outil affiché directement dans la sidebar⁷ sur le bureau Windows. Ces petits modules sont, en général, connectés en permanence avec un serveur distant. Ils permettent donc l'affichage d'informations **en temps réel**. D'un point de vue technologique, un gadget utilise principalement le HTML, les styles CSS et le JavaScript.

Un gadget Windows est en réalité une archive compressée avec l'extension ".gadget". Afin de créer un gadget fonctionnel, cette archive doit au minimum contenir les fichiers suivants :

- ❑ Le fichier "**gadget.XML**" contient toutes les informations concernant le gadget comme par exemple le nom, la description, l'auteur et la version.
- ❑ Le fichier "**settings.html**" qui permet de paramétrer le gadget si celui-ci en a besoin.
- ❑ Un fichier "***.png**" qui représente l'image du gadget dans la boîte à gadget accessible en faisant clic droit sur le bureau, puis "Gadget".
- ❑ Un fichier "***.html**" qui représente la structure et l'apparence général du gadget.

Un gadget possède en général deux états : docked et undocked. Ces deux états représentent la version grande (undocked) et la version minimisée (docked). Afin de passer d'un état à l'autre, il suffit de cliquer sur l'icône représentée par une flèche, située à droite du gadget.

Si ces deux états ne suffisent pas à la réalisation du gadget, il est possible d'utiliser des éléments appelés "flyout". Ces flyouts sont des fichiers html qui s'ouvrent sur la partie gauche du gadget. Ils permettent l'affichage d'informations supplémentaires. Attention, un seul flyout peut s'afficher à la fois !

5.2 Fonctionnalités

5.2.1 Vérification de l'état du serveur

Cette fonctionnalité permet de détecter si le serveur d'ouschool.ch est accessible ou pas. Soit le serveur Web ne fonctionne plus temporairement pour maintenance ou autre, soit l'utilisateur ne dispose d'aucune connexion à internet. Si c'est le cas, le gadget ne sera pas utilisable et un message d'erreur s'affichera.



Ecran 12 - Vérification connexion serveur

⁷ Partie droite du bureau Windows prévue pour les gadgets.

5.2.2 Authentification

Un système d'authentification est nécessaire pour pouvoir, par la suite, y afficher les informations concernant cet utilisateur. L'option la plus appropriée pour réaliser cette fonctionnalité est d'utiliser le module "Réglages" symbolisé par la clé à molette. Ce module existe sur la plupart de gadget Windows. Il permet de stocker des informations qui seront réutilisables à travers tout le gadget. Ce sont des objets de type "System.Gadget.Settings" qui pourraient être comparés à des cookies.

Les informations demandées dans cette partie sont l'adresse email ainsi que le mot de passe. Ces informations stockées dans des objets systèmes seront par la suite utilisés pour la connexion au Web service privé. Si l'utilisateur rentre des données erronées ou que son compte n'ait pas encore été activé, un message d'erreur s'affiche.



Ecran 13 - Fenêtre de login



Ecran 14 - Réglages utilisateur incorrects

5.2.3 Visualisation des prochaines réservations

Une fois l'authentification réussie, l'utilisateur peut utiliser toutes les fonctionnalités du gadget. Sur la version minimisée (docked) du gadget, s'affichent les réservations d'un bien de l'utilisateur pour aujourd'hui ainsi que celles des sept prochains jours. Un bouton servant à la gestion des réservations de tous les biens de cet utilisateur est également présent sur cette version du gadget.



Ecran 15 - Affichage réduit (docked)

5.2.4 Affichage dynamique de nouvelles réservations

Une des fonctionnalités principales d'un gadget est d'obtenir **en temps réel** des informations. Afin d'ajouter une plus-value au gadget, une étoile s'affiche automatiquement en-haut à droite du gadget quand une nouvelle réservation d'un bien a été faite. De plus, à l'intérieur, le nombre de nouvelles réservations s'affiche.



Ecran 16 - Notification de nouvelles réservations

Lorsque l'utilisateur clique sur cette étoile, les nouvelles réservations s'affichent dans un flyout. La hauteur de cette fenêtre s'adapte au nombre de réservation. L'utilisateur peut ainsi prendre une décision concernant chaque demande de réservation :

- ❑ Accepter à l'aide de l'icône vert
- ❑ Refuser à l'aide de la croix rouge



Ecran 17 - Affichage des nouvelles réservations

Une troisième option s'affiche au niveau de chaque nouvelle réservation. L'utilisateur a la possibilité de voir le détail de la réservation. Ces détails contiennent principalement les coordonnées de la personne ayant effectué la réservation. Cette option n'est pas accessible seulement depuis cette page mais également depuis toutes les pages où est affichée une quelconque réservation.

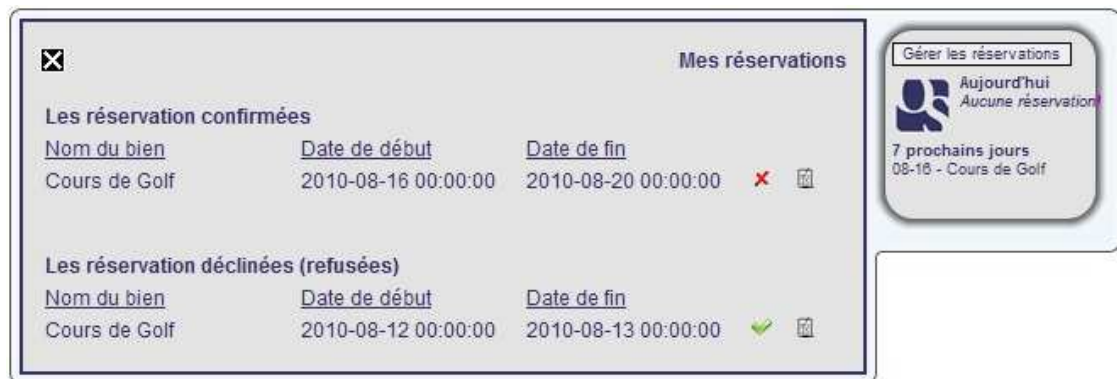


Ecran 18 - Affichage du détail d'une réservation

5.2.5 Gestion des réservations

Le choix fait sur une réservation peut en tout temps être modifié. Pour se faire, il suffit de cliquer sur le bouton situé au sommet du gadget. Une liste avec toutes les réservations futures s'affiche qui adapte sa hauteur au nombre d'enregistrement. Le

prestataire a alors la possibilité de changer de choix. Depuis cet endroit, l'utilisateur peut également voir le détail de chaque réservation.



Ecran 19 - Gestion des réservations

5.2.6 Gestion des biens/services

Passons maintenant à la grande version (undocked) du gadget. Les informations de la petite version y restent affichées. C'est à la suite de ces informations que la partie concernant la gestion des biens / services est insérée. L'utilisateur peut depuis cette fenêtre gérer tous ces biens.

- ☐ Ajouter un bien à l'aide du bouton prévu à cet effet
- ☐ Gérer les disponibilités à l'aide du 1^{er} icône (calendrier)
- ☐ Modifier un bien à l'aide du 2^{ème} icône (stylo)
- ☐ Supprimer un bien à l'aide du 3^{ème} icône (croix)



Ecran 20 - Version grande (undocked)

5.3 Ajouter / modifier un bien ou un service

Les formulaires permettant d'ajouter et de modifier des biens sont exactement les mêmes à la seule différence que lorsque l'utilisateur modifie son bien, les champs sont déjà pré-remplis à l'aide du Web service.

L'affichage de ce formulaire se fait dynamiquement, mis à part la liste déroulante qui est affichée pour tous les types de biens. Etant donné que l'application se veut

générique, les différents champs du formulaire changent automatiquement selon le type de services que le prestataire de service propose (hébergement, moniteur, transport et matériel). Si l'équipe outhschool désire rajouter une catégorie, il suffit de se connecter sur la zone d'administration et d'ajouter une catégorie. Une fois la catégorie créée, il faut encore aller rajouter des champs pour cette catégorie toujours via la zone d'administration de la plateforme Web. Le prochain partenaire qui choisira cette nouvelle catégorie aura un formulaire différent de celui ci-dessous, il y aura seulement les champs qui ont été rajoutés à cette catégorie.

De plus, lorsque l'équipe outhschool rajoute un champ, il doit également choisir un type de champs. Dans le cadre de ce projet, trois types de champs peuvent être choisis et seront affichés différemment dans le formulaire.

Type de champs	Code HTML	Exemple (voir ci-dessous)
Text	<code><input type="Text"/></code>	<i>Activité dispensée</i>
Texte long	<code><textarea></textarea></code>	<i>Description</i>
Boolean	<code><input type="Checkbox"/></code>	<i>Langue parlée</i>

Dans le cadre de ce projet, aucun champ n'est obligatoire. Mais l'insertion de code HTML dans un élément du formulaire est totalement interdit.

Ecran 21 - Ajout et modification d'un enregistrement

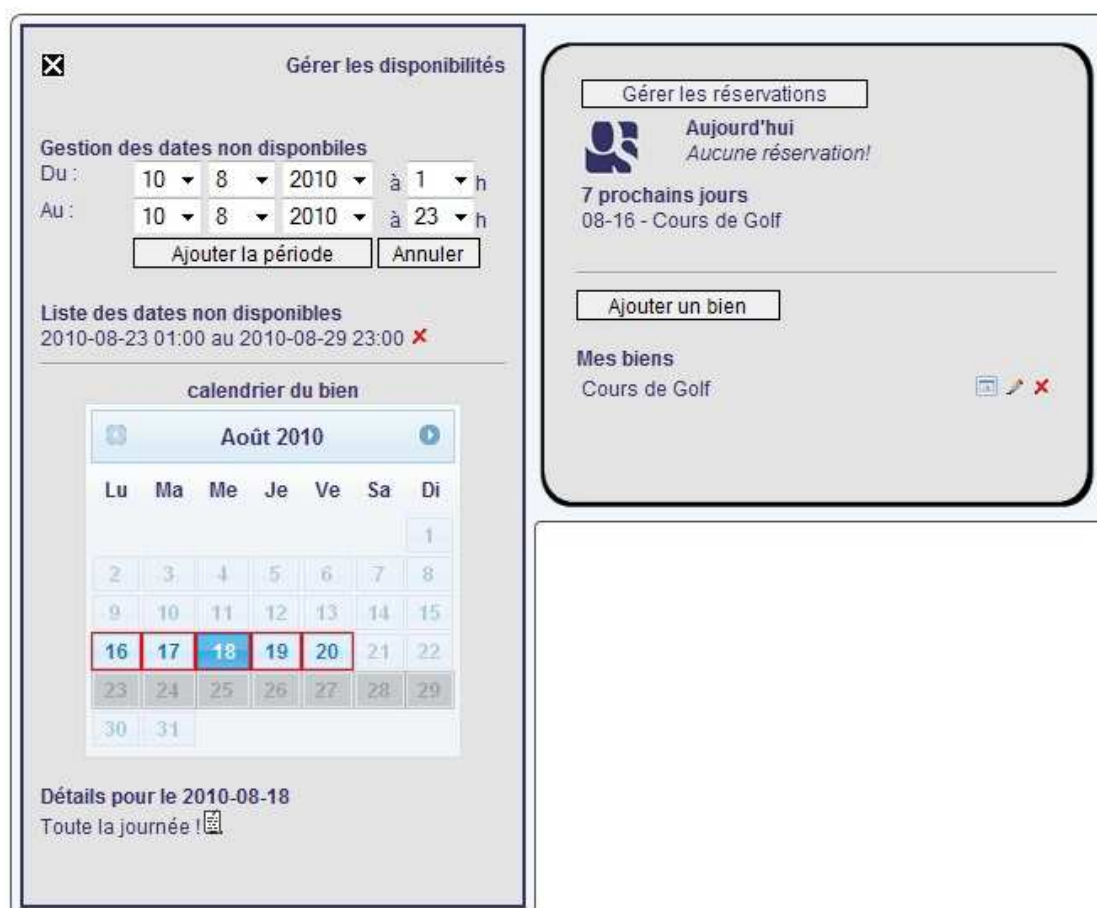
5.4 Gestion des disponibilités

La partie concernant les disponibilités d'un bien s'affiche comme la plupart des autres fenêtres dans le flyout du gadget. À cet endroit, l'utilisateur a la possibilité d'effectuer plusieurs actions.

- ❑ **Bloquer certaines dates** auxquelles le bien n'est pas disponible. Par exemple, pour un prestataire qui propose un hébergement, s'il reçoit une réservation par téléphone (et non pas depuis la plateforme outhschool), il peut aller directement dans son gadget et ainsi bloquer la plage de dates réservée par téléphone. Cet hébergement ne sera alors plus proposé lors d'une

réserveation sur la plateforme. Les plages de dates bloquées apparaissent alors grisées sur le calendrier.

- ❑ **Avoir un aperçu des jours libres/occupés** sur un calendrier. Cela permet facilement de voir quels jours sont occupés car ils sont encadrés en rouge sur le calendrier. De plus, l'utilisateur a la possibilité de cliquer sur les jours où une réserveation est faite. Au fond de la fenêtre s'affiche le détail de la journée sélectionnée. Depuis cet endroit, l'utilisateur peut également visionner le détail de la réserveation.



Ecran 22 - Gestion des disponibilités d'un enregistrement

5.5 Technologies utilisées

5.5.1 HTML et CSS

Les principaux langages pour la mise en page d'un gadget Windows sont le **HTML** et les styles **CSS**. Ces deux technologies sont les 2 principaux langages Web permettant la mise en page de site Web. Ils sont donc également utilisés dans les gadgets Windows.

Etant donnée que pour la petite version (docked) et la grande version (undocked) du gadget, le fichier **HTML** de source est le même, c'est l'attribut "visibility" des styles **CSS** qui permet de cacher certaines parties entre les deux versions. Le JavaScript intervient également dans la mise en page des deux versions, car c'est à travers de ce langage que la taille de la police est modifiée selon la version affichée à l'écran.

Les différents formulaires présents dans ce gadget sont totalement réalisés à l'aide de balises **HTML** (input, form, table). Le traitement de ces derniers se fait en JavaScript. L'événement "onClick" permet de récupérer toutes les valeurs du formulaire avant de les envoyer sur le serveur distant.

5.5.2 Javascript avec l'objet XMLHttpRequest

Afin de permettre la communication entre le gadget Windows et Web service, il a fallu faire appel à l'objet JavaScript **XMLHttpRequest**. Cet objet permet d'obtenir des données sous différents formats (XML, JSON ou encore du texte simple) à l'aide de requêtes HTTP exécutées sur un serveur distant.

Pour chaque requête effectuée sur le serveur distant, un objet de ce type est créé. Afin de pouvoir établir la connexion, l'objet fait appel à la fonction **open()**. Cette fonction prend plusieurs éléments en paramètre. Les deux derniers paramètres permettent de se connecter sur un serveur protégé par une authentification.

- ❑ **La méthode HTTP** désirée
 - GET Lecture de données
 - POST Ajout de données
 - PUT Modification de données
 - DELETE Suppression de données
- ❑ **L'URL** du serveur distant à atteindre
- ❑ **"True" ou "False"** si l'on fait un appel synchrone ou asynchrone. La différence entre ces deux appels est que, de façon synchrone, le script va attendre la réponse du serveur avant de passer à l'étape suivante. Tandis que de façon asynchrone, il n'attend pas cette réponse.
- ❑ (optionnel) **Le nom d'utilisateur**
- ❑ (optionnel) **Le mot de passe**

A la suite de cette méthode, c'est la fonction **send()** qui est appelée. Si la méthode définit en paramètre numéro 1 de la fonction open est "GET", la fonction **send()** ne prend aucun élément en paramètre. Dans le cas contraire, si l'utilisateur utilise une autre méthode qui affecte les données distantes, un paramètre ayant la même structure qu'un objet distant doit être passé dans cette fonction. Cela peut très bien être un "string" qui contient une structure XML si l'URL distante demande du XML bien sûr.

Pour finir, le résultat de la requête envoyée est stocké dans un autre attribut. Cet attribut peut faire appel à plusieurs propriétés permettant d'obtenir le résultat de la requête.

- ❑ **ResponseText** retourne le résultat sous forme de chaîne de caractères
- ❑ **ResponseXML** retourne le résultat sous forme d'objet XML
- ❑ **Status** permet de retourner le code HTML de la requête
 - 200 OK
 - 400 Erreur de syntaxe dans la requête
 - 401 Non-autorisé (lorsqu'une authentification est nécessaire)
 - 404 Page introuvable
 - 405 Méthodes (GET, POST...) pas autorisées
 - 500 Erreur interne au serveur

5.5.3 Librairie JQuery

JQuery est une librairie Javascript gratuite. Cette librairie est composée d'un seul fichier Javascript qui doit être lié au fichier HTML faisant appel à cette dernière. Cette librairie facilite l'intégration de Javascript à l'intérieur d'une page Web. Elle contient notamment les fonctionnalités suivantes :

- ❑ Gestion des événements (auto-complete d'une liste déroulante...)
- ❑ Affichage de messages dynamiques avec un bon rendu visuel
- ❑ Intégration d'élément AJAX

Fiche technique	
Année de parution	2006
Développement	John Resig
Version actuelle	1.4.2
Licence	GNU GPL
Site internet	www.jquery.com

6 Problèmes rencontrés

6.1 *Délai du cahier des charges*

Étant donné que l'activité des premières semaines s'élevait à 50%, le délai de deux semaines pour rendre le cahier des charges ainsi que les différents documents demandés était très court. Des heures supplémentaires ont dû être faites pour rendre les différents éléments dans le temps.

6.2 *Nouvelles technologies*

6.2.1 *Nouveau langage*

La méconnaissance du langage de développement a posé beaucoup de problèmes pour la réalisation de ce travail. Comme tout apprentissage de nouveaux langages, il faut de la patience et surtout beaucoup de motivation. Une période d'apprentissage a été nécessaire pour assimiler les bases du fonctionnement du langage.

Durant tout le projet, de nombreuses recherches ont été effectuées sur la toile afin de répondre aux questions sans réponse. Certains problèmes ont également été exposés sur des forums d'entraide afin de pouvoir les résoudre le plus rapidement possible et éviter de perdre trop de temps.

6.2.2 *Nouvel environnement de développement*

L'installation du framework et plus particulièrement la mise en place de la communication entre le système de gestion de base de données MySQL et les applications du framework a été très laborieuse. Au début, différents tutoriaux ont été suivis pour faire cette connexion. Tous proposaient l'opération en ligne de commande et un problème inconnu m'empêchait de faire cette installation correctement. Après plusieurs jours de recherches, un fichier exécutable a été trouvé sur internet et a permis en un seul clic de résoudre ce problème.

6.2.3 *API pour les requêtes SQL*

Le framework propose une API qui permet de remplacer les requêtes SQL. Un temps d'adaptation a été nécessaire et encore maintenant cette manière d'interroger la base de données ne peut pas être considérée comme acquise.

6.2.4 *Configuration de l'envoi d'email*

Après avoir essayé d'installer un serveur SMTP⁸ en local, l'envoi d'email ne fonctionnait pas. Après plusieurs essais avec différents serveurs SMTP distants, c'est le serveur de Gmail qui a été utilisé pour la plateforme de développement. Une fois l'application hébergée sur le serveur de production, les informations pour l'envoi de courriels feront appel au serveur SMTP local.

6.2.5 *Récupération du projet existant*

Le fait de devoir récupérer une application existante a posé quelques problèmes. Plusieurs plugins utilisés par le développeur de la version 1.0 n'avaient pas été installés sur mon serveur local, ce qui a provoqué plusieurs erreurs lors de la compilation du projet. De plus, après installation de ces plugins manquants, des erreurs apparaissaient toujours. Après une profonde réflexion, le problème a été

⁸ Protocole permettant l'envoi de courriels

résolu : la version des plugins installée en local n'était pas la même que celle utilisée par le développeur de la version 1.0 de la plateforme. L'installation de ces plugins manquants m'a permis de découvrir un plugin très utile. Il s'agit d'un outil de débogage très performant.

6.3 *Communication avec groups.ch*

Au début du projet et selon l'équipe outhschool, la plateforme Web groups.ch devait nous donner une copie de leur base de données afin de pouvoir l'installer en local et ainsi tester l'intégration d'une base de données externe sur la plateforme d'outhschool. Finalement l'acquisition de cette base de données n'a pas été si simple que ça.

Dans un premier temps, il a fallu leur envoyer un dossier complet avec l'architecture de l'application et une explication claire et précise de la future utilisation de leurs données. Une fois ce dossier envoyé, aucune réponse de leur part n'a été reçue.

Mi-juillet, n'ayant toujours pas eu de réponse de leur part, j'ai appelé directement groups pour savoir où en était l'analyse de mon dossier. Grande surprise, la directrice m'a dit qu'elle n'avait plus reçue de mail de ma part depuis mi-juin. Je leur ai donc à nouveau envoyé le dossier. Quatre heures plus tard, la réponse est tombée: groups.ch ne veut pas fournir une copie de leur base de données.

Dès ce moment, il a fallu réfléchir à une alternative pour palier à ce gros problème. Après réflexion, deux possibilités ont été faites au professeur responsable du suivi du projet.

- ❑ Soit modéliser une base de données ressemblant à celle de groups.ch en se basant sur leur site internet.
- ❑ Soit utiliser OsCommerce, un outil open source permettant de créer sa boutique en ligne. Avec ce système, l'idée était de définir une catégorie qui serait "location" et ensuite utiliser les éléments de cette catégorie pour les mapper avec la plateforme outhschool.

La deuxième solution a été plus appréciée par le professeur. Mais après une dernière tentative de ce dernier pour l'obtention d'une copie de la base de données, groups.ch a finalement donné une copie de leur base de données.

Après analyse de cette base de données, j'ai remarqué que ce n'était certainement pas la vraie base de données. Car les champs "Locality" et "Country" était directement intégré dans la table des établissements. Il n'y avait pas de table à part pour les pays et les localités. Finalement, cette base de données a été utilisée pour l'affichage des informations dans le formulaire de réservation.

6.4 *Communication entre les applications*

Au début du projet, je n'avais aucune idée en développement avec le framework django. Je ne connaissais également rien des gadgets Windows. Donc la manière pour relier ces deux applications était vraiment loin d'être définie.

Après quelques recherches et les premières applications développées, la situation s'est peu à peu éclaircie et la manière de communiquer entre ces deux applications est naturellement venue avec la mise en place d'un Web service.

7 Améliorations

Il reste encore du travail du côté de la zone d'administration de la plateforme Web. L'outil de mapping entre les tables des bases de données externes et celle d'outhschool doit encore être développé. Le temps pour ce travail est estimé à 40 heures.

Au niveau du gadget Windows, l'aspect ergonomique peut certainement être amélioré. Une fonctionnalité supplémentaire pourrait être ajoutée à cet outil : une synchronisation des disponibilités avec "Google Calendar". Cette option serait certainement très utile pour les personnes proposant des services de type "Moniteur". Pour les autres catégories, cette option n'aurait aucun intérêt.

Egalement au niveau du gadget, il serait envisageable de programmer une version asynchrone de l'outil. Ce qui permettra d'utiliser le gadget même si le serveur n'est pas atteignable. Une sauvegarde des données enregistrées serait faite à chaque action de l'utilisateur. Une fois la connexion rétablie, ces données seraient synchronisées avec la base de données.

Etant donné que le seul outil de gestion de contenu est un gadget utilisable uniquement sous Windows, il serait utile de développer un autre outil pour la gestion des biens. L'équipe outhschool a déjà demandé de réaliser un gadget pour Mac OS⁹. Ceci ne devrait pas poser de problème étant donné que l'architecture incluant le Web service est déjà en place et fonctionnel. Dans le même sens, la création d'un module pour un CMS de type Joomla peut être très utile. Cette option permettra à l'utilisateur de pouvoir gérer le contenu de la plateforme outhschool directement depuis la partie d'administration de son propre site Web.

Ce qui serait intéressant au niveau du formulaire de réservation sur la plateforme Web, c'est la génération automatique du budget. La base de données a déjà été prévue pour cette fonction. Cette fonction disposerait d'un lien sur la page de résumé qui permettra de télécharger le budget au format PDF. A l'intérieur de ce fichier, le prix détaillé de chaque service choisi serait inscrit ainsi que le montant total du camp entier.

Au niveau de l'intégration des partenaires, il serait bien d'ajouter la possibilité de lier un Web service au système d'informations d'outhschool. Actuellement seule une base de données peut être liée. Les accès à une base de données sont difficilement accessibles, la solution d'un Web service est peut être la bonne pour palier à ce problème.

⁹ Système d'exploitation utilisé sur les ordinateurs de type Apple

8 Gestion de projet




















Un cahier des charges a été défini lors des deux premières semaines de projet. Ce dernier a été signé et approuvé par l'équipe outhschool.ch ainsi que le professeur responsable de ce travail.






8.1 Technique de gestion de projet




Une planification MS-Projet a été mise en place en début de projet afin de définir les différentes tâches ainsi que la durée de chaque tâche. Ce qui a permis de faire le suivi des tâches via ce programme. La fiche détaillée se trouve en annexe.

8.2 Fonctionnalités réalisées

Ci-dessous se trouve une liste de toutes les fonctionnalités obligatoires et optionnelles qui ont été réalisées dans le cadre de ce travail. Le terme "Gestion" dans ce tableau fait référence aux actions (ajout, modification, suppression et éventuellement la validation) sur un élément.

Fonctionnalités obligatoires	
Architecture	
Création d'un Web service	
Modélisation de la base de données	
Plateforme Web	
Formulaire d'inscription	
Validation d'une inscription	
Insertion des réservations dans la base de données	
Module de recherche multicritère	
Formulaire de réservation	 ~90%
Affiche d'enregistrements de base de données externes	
Suivi d'une réservation (état en attente, confirmé...)	
Interface d'administration	
Intégration d'une base de données externe	 ~10%
Gestion des partenaires	
Gestion des réservations	
Gestion des biens/service des prestataires de services	
Création dynamique de package	
Gadget Windows	
Accès au Web service (lecture – écriture)	
Paramétrage de l'outil	
Aperçu des disponibilités sur un calendrier	
Gestion des biens de l'utilisateur	
Formulaire d'ajout et de modification générique	

Fonctionnalités optionnelles	
Autre	
Création d'un module de gestion intégré à un CMS	
Intégration des moniteurs J+S du canton de fribourg	
Gadget Windows	
Gestion des réservations	
Affichage dynamique de nouvelles réservations	
Asynchronisme de l'outil	

Légende	
	Fonctionnalités terminées à 100%
 %	Fonctionnalités débutées
	Fonctionnalités pas commencées

8.3 *Ecart de temps entre prévu et réalisé*

Tout le monde sait qu'il est très difficile d'estimer le nombre d'heures qu'une tâche va prendre. Mais comme souvent, des écarts proviennent suite à des imprévus. Suite au retard accumulé pour la livraison de la base de données de groups.ch, beaucoup plus de temps a été consacré à la programmation du gadget Windows que pour la partie Web avec le formulaire de réservation en ligne. Des écarts entre les heures prévues et celles réalisées sont plus réduits concernant des tâches moins importantes. Ils sont disponibles ci-dessous.

Tâche	Nombre d'heures prévues	Nombre d'heures réalisées
Gadget Windows	48	100
Formulaire de reservation	12	30
Formulaire d'inscription	10	24
Création du Web service	20	30
Zone d'administration	64	20
Formulaire de recherche multicritère	4	10
Création des classes métiers	9	11
Rédaction du document final	40	50

8.4 *Déroulement du projet*

Ce projet s'est déroulé pendant 11 semaines du 17 mai 2010 au 16 août 2010. Un rapport hebdomadaire a également été remis au professeur responsable à la fin de chaque semaine de travail. La période du 21 juin au 3 juillet a été consacrée à la préparation des examens de module ainsi qu'à la réalisation de ces derniers. Le projet s'est décomposé en différentes parties visibles ci-dessous.

Période	Tâches
17.05.2010 – 30.05.2010	<input type="checkbox"/> Prise de connaissance du projet <input type="checkbox"/> Cahier des charges <input type="checkbox"/> Diagramme de USE-CASE
31.05.2010 – 16.06.2010	<input type="checkbox"/> Apprentissage Python & Django <input type="checkbox"/> Prise en main de la version 1.0
17.06.2010 – 20.06.2010	<input type="checkbox"/> Diagramme de classes
21.06.2010 – 03.07.2010	<i>Préparation et examens de module</i>
04.07.2010 – 07.07.2010	<input type="checkbox"/> Classes métiers <input type="checkbox"/> Base de données <input type="checkbox"/> Prototypes écran
08.07.2010 – 19.07.2010	<input type="checkbox"/> Formulaire d'inscription <input type="checkbox"/> Web service
20.07.2010 – 31.07.2010	<input type="checkbox"/> Gadget Windows
01.08.2010 – 06.08.2010	<input type="checkbox"/> Formulaire de réservation
07.08.2010 – 15.08.2010	<input type="checkbox"/> Documentation

8.5 Nombre d'heures réalisées

Le nombre d'heures total cumulées pour la réalisation de ce projet s'élève à **363 heures**. En annexe se trouve un décompte détaillé des heures passées sur ce projet chaque jour pendant ces deux mois.

9 Conclusion

9.1 Généralités sur le projet

En conclusion, le projet s'est dans l'ensemble très bien déroulé. Environ 90% des fonctionnalités planifiées ont été réalisées. Tous les délais de restitution des différents documents (rapports hebdomadaires, cahier des charges, rapport final) ont été respectés à la lettre.

Malheureusement, suite à des problèmes techniques et une collaboration plus difficile que prévu avec groups.ch, leur base de données a été délivrée très tard dans le projet. Ce qui a empêché partiellement la mise en place de la fonctionnalité d'intégration de base de données externes.

Le fait de planifier toutes les tâches à l'avance n'était certainement pas la meilleure solution. Ne connaissant pas les technologies, il est encore plus difficile d'estimer le temps que va prendre une fonctionnalité. Une méthodologie dans la gestion de projet de type "**scrum**" aurait certainement été plus appropriée avec des itérations d'une semaine. Ces itérations auraient permis de définir plus précisément les tâches semaine par semaine.

9.2 D'un point de vue personnel

Personnellement, ce projet m'a beaucoup apporté. Le fait de devoir comprendre et résoudre des problèmes de manière autonome m'a demandé beaucoup de patience et d'implication personnelle. Je n'avais jamais vécu telle situation et cela m'a certainement bien préparé à mon entrée dans la vie professionnelle.

Le fait d'avoir appris un nouveau langage de programmation Web n'a été que bénéfique pour moi. Cela représente une compétence supplémentaire à faire apparaître dans mon curriculum vitae. Par ailleurs et malgré les difficultés rencontrées en début de projet, j'ai finalement beaucoup apprécié cette nouvelle manière de développer des sites Web. Certainement que dans les développements futurs, le framework django sera prioritaire sur le langage PHP. Ce qui n'était pas le cas avant ce projet. De plus, ce projet m'a démontré qu'il n'existe pas qu'un seul langage de programmation et qu'il ne faut pas se reposer sur ses acquis. Il ne faut pas hésiter à apprendre des nouveaux langages car certains d'entre eux sont plus performants que ce qu'on peut croire.

Etant donné que dans un futur proche je veux travailler dans le domaine du développement d'applications Web, ce travail m'a permis de confirmer mon chemin futur.

10 Tables des illustrations

10.1 Table des schémas

Schéma 1 - Sites Web statiques	4
Schéma 2 - Sites Web dynamiques	5
Schéma 3 - Architecture de l'application	15
Schéma 4 - Diagramme de classes.....	16
Schéma 5 - Fonctionnement du Web service	25

10.2 Table des extraits de code

Extrait de code 1 - Exemple XML	11
Extrait de code 2 - Exemple JSON.....	11
Extrait de code 3 - Exemple YAML.....	12

10.3 Table des écrans

Ecran 1 - Formulaire de réservation (version 1.0)	14
Ecran 2 - Formulaire de recherche multicritère.....	17
Ecran 3 - Formulaire d'inscription.....	18
Ecran 4 - Courriel de confirmation.....	19
Ecran 5 - Formulaire de modification de mot de passe.....	19
Ecran 6 - Formulaire d'informations sur la base de données	20
Ecran 7 - Formulaire de réservation → Saisie des informations principales	21
Ecran 8 - Formulaire de réservation → Sélection des services.....	22
Ecran 9 - Formulaire de réservation → Coordonnées de contact	23
Ecran 10 - Formulaire de réservation → Confirmation.....	23
Ecran 11 - Suivi d'une réservation.....	24
Ecran 12 - Vérification connexion serveur	28
Ecran 13 - Fenêtre de login.....	29
Ecran 14 - Réglages utilisateur incorrects	29
Ecran 15 - Affichage réduit (docked)	29
Ecran 16 - Notification de nouvelles réservations	30
Ecran 17 - Affichage des nouvelles réservations.....	30
Ecran 18 - Affichage du détail d'une réservation	30
Ecran 19 - Gestion des réservations	31
Ecran 20 - Version grande (undocked).....	31
Ecran 21 - Ajout et modification d'un enregistrement.....	32
Ecran 22 - Gestion des disponibilités d'un enregistrement	33

11 Sources

N°	Détail de la source
#1	Titre : Le langage PHP - Tutorial PHP-MySQL Auteur : Van Lancker Luc http://www.lehtml.com/php/php1.html
#2	Titre : MySQL - PostgreSQL? Des arguments pour comprendre Auteur : Antoine Crochet-Damais http://www.journaldunet.com/solutions/0402/040210_postgresql.shtml
#3	Titre : XMLHttpRequest Auteur : Robloche http://www.toutjavascript.com/savoir/XMLhttprequest.php3
#4	Titre : Webservice Auteur : Academic télécommunication center http://fr.academic.ru/dic.nsf/frwiki/1729732
#5	Titre : Le XML expliqué Auteur : Tetraedre Company http://www.tetraedre.com/advanced/XML.php
#6	Titre : Création d'un gadget pour la Windows Sidebar Auteur : MORAND Louis-Guillaume http://lqmorand.developpez.com/articles/sidebar-gadget/
#7	Titre : Communauté django francophone http://www.django-fr.org/
#8	Titre : Introduction à XML http://www.commentcamarche.net/contents/XML/XMLintro.php3
#9	Titre : YAML Ain't Markup Language http://www.yaml.org/start.html
#10	Titre : TIOBE Software: The Coding Standards Company http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html
#11	Titre : Page officielle du composant Django-rest-interface http://code.google.com/p/django-rest-interface/
#12	Titre : Page officielle du composant Django-piston http://bitbucket.org/jesperi/django-piston/wiki/Home
#13	Titre : Site officiel du projet Django (en anglais) http://www.djangoproject.com/
#14	Titre : Site officiel de l'IDE eclipse http://www.eclipse.org/
#15	Titre : Site officiel du SGBD MySQL http://www.mysql.org/

#16	Titre : Site officiel du langage JSON http://www.json.org/
#17	Titre : Site officiel de la librairie Javascript JQuery http://jquery.com/
#18	Titre : Site officiel de Microsoft http://www.microsoft.com
#19	Titre : Wikipedia → L'état de l'art http://fr.wikipedia.org/wiki/%C3%89tat_de_l%27art
#20	Titre : Wikipedia → Langage Java Server Pages http://fr.wikipedia.org/wiki/JavaServer_Pages
#21	Titre : Wikipedia → Langage Python http://fr.wikipedia.org/wiki/Python_%28langage%29
#22	Titre : Wikipedia → Langage PERL http://fr.wikipedia.org/wiki/Perl_%28langage%29
#23	Titre : Wikipedia → Langage PHP http://fr.wikipedia.org/wiki/PHP#cite_note-manpreface-2
#24	Titre : Wikipedia → ColdFusion http://fr.wikipedia.org/wiki/ColdFusion
#25	Titre : Wikipedia → Librairie JQuery http://fr.wikipedia.org/wiki/JQuery

12 Annexes

12.1 *Cahier des charges*

Afin d'automatiser le processus de réservation, les fonctionnalités suivantes devront être réalisées durant ce travail de bachelor.

12.1.1 Architecture

- ❑ Création d'un Web service
- ❑ Création d'un module permettant la synchronisation des bases de données des partenaires avec celle d'outhschool.ch
- ❑ Création de la base de données d'outhschool.ch

12.1.2 Fonctionnalités sur l'existant

- ❑ Modifier le module de réservations personnalisées
 - Insertion des demandes de réservation dans la base de données
 - Intégration des offres des partenaires depuis leur base de données
 - Intégration des offres des prestataires de services individuels
- ❑ Modifier le module de réservation d'un package "All-inclusive"
 - Insertion des demandes de réservation dans la base de données
 - Module de recherche multicritères sur les packages

12.1.3 Fonctionnalités pour l'équipe outhschool.ch

- ❑ Création dynamique de package à l'aide des offres des partenaires et des prestataires de services.
- ❑ Gestion des demandes de réservation
 - Validation avec état : en attente, en traitement, payé
 - Modification d'une réservation
 - Annulation/suppression d'une réservation
- ❑ Gestion des partenaires et des prestataires de services
 - Confirmer la demande de collaboration
 - Insérer un partenaire
 - Modifier un partenaire / prestataire de services
 - Donner les accès et les outils (widget Windows)

- ❑ Intégration de partenaires et plus particulièrement de leurs bases de données via un module de synchronisation

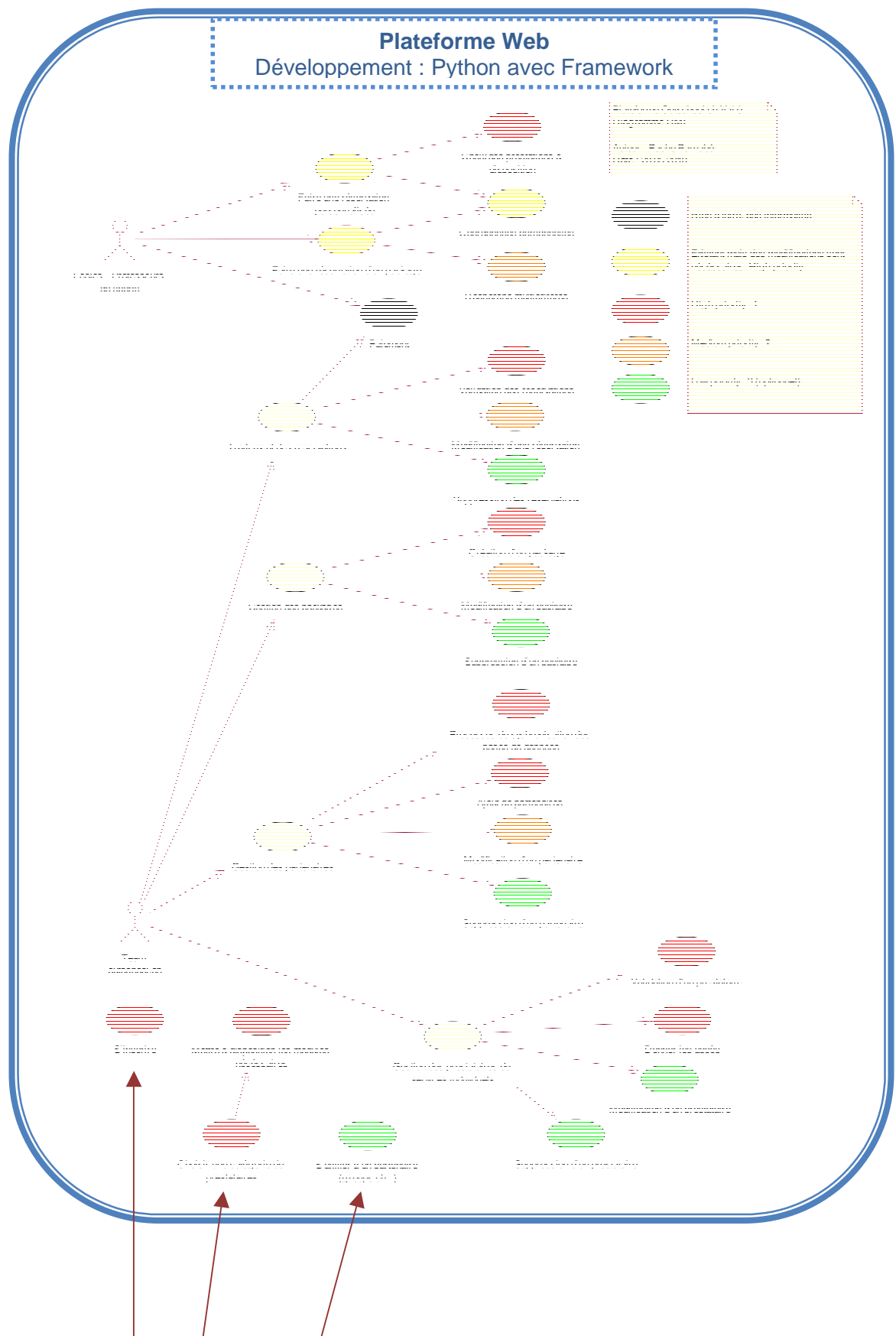
12.1.4 Fonctionnalités concernant les prestataires de services

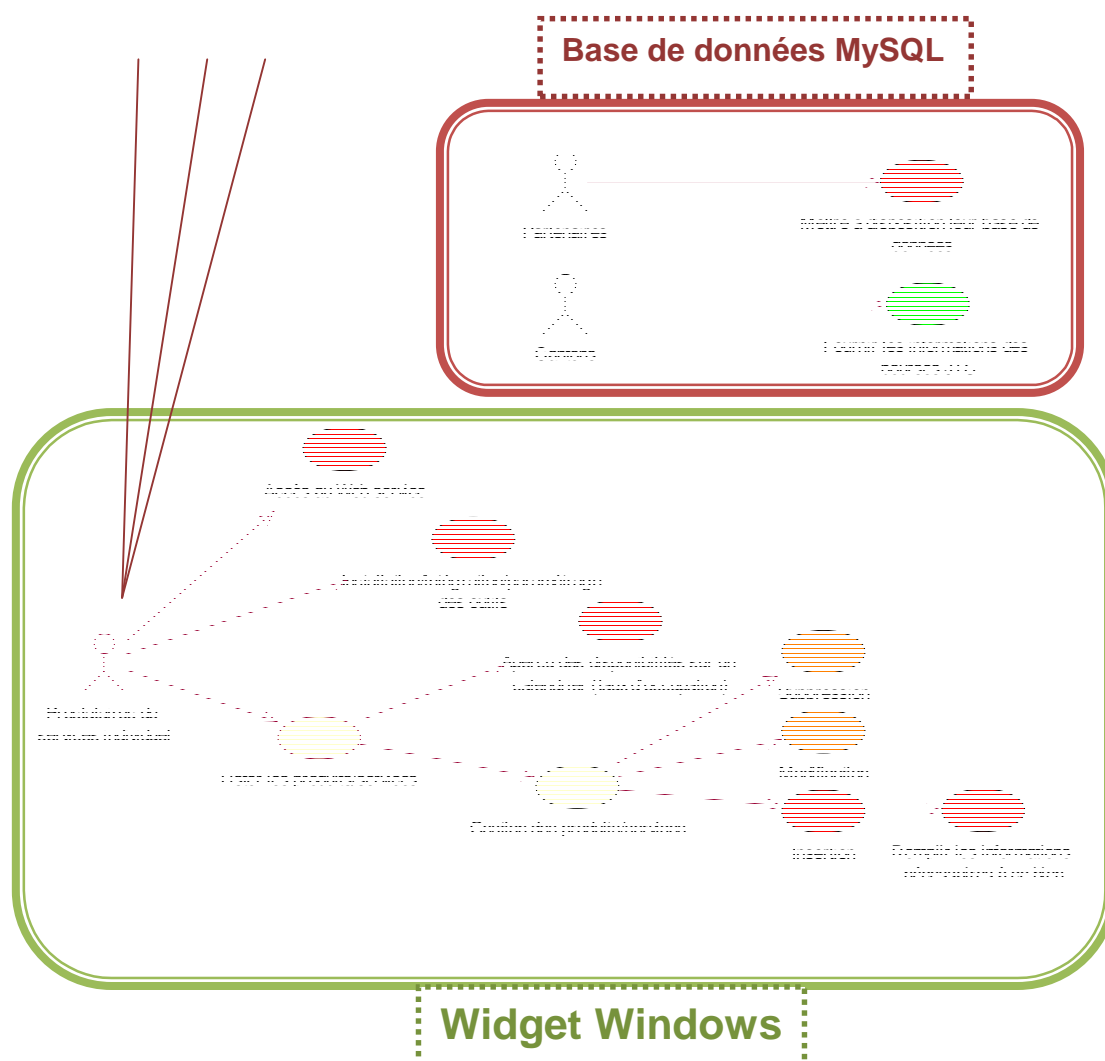
- ❑ Formulaire de collaboration/d'inscription
- ❑ Possibilité de s'affilier à un partenaire
- ❑ Widget Windows 7 avec la liste de tous les produits/services
 - Ajout + modification
 - Personnalisation des modules par rapport au type de produits/services (logement, professeur J+S, matériel de sport etc...).
 - Aperçu par mois du taux d'occupation sur un calendrier (plage de date grisé si non disponible)

12.1.5 Fonctionnalités optionnelles

- ❑ Suppression des différents enregistrements dans la base de données via l'interface d'administration.
- ❑ **Asynchronisme du widget Windows 7** : Cela signifie que le module fonction également hors-ligne et stocke provisoirement les données en local.
- ❑ Intégration de la liste des professeurs Jeunesse & Sport provenant de la bourse aux moniteurs du site du canton de fribourg.
- ❑ **Création d'un module de gestion pour les CMS** : Cela signifie que le prestataire de services pourra intégrer directement la gestion de ses biens dans le CMS de son site.
 - Module joomla
 - Module wordpress

12.2 Diagramme de Use-Case





12.3 Comment reprendre le projet

Cette annexe aide toute personne à reprendre le projet outschool.ch version 1.1. On part du principe que eclipse est déjà téléchargé et fonctionnel. Toutes les ressources se trouvent sur le CD-ROM ci-joint dans le dossier "Outils".

12.3.1 Installer Python

- ❑ Python-2.6.5.exe

12.3.2 Installer django

- ❑ Décompresser Django-1.1.1.tar
- ❑ Ouvrir l'invite de commande : Démarrer → Exécuter → cmd
- ❑ Se rendre dans le dossier précédemment décompressé "Django-1.1.1" à l'aide de la commande "cd PathToFolder"
- ❑ Taper "setup.py install" → Enter

12.3.3 Installer les composants nécessaires

Ces composants s'installent de la même manière que django (point 12.4.2)

- ❑ Liste des fichiers à décompresser
 - dcramer-django-debug-toolbar-a0913fd.zip
 - digi604-django-cms-2.0-2.0.2-0-gc0288a1.zip
 - django-mptt-0.2.1.zip
 - django-reversion-1.2.1.tar
- ❑ Liste de l'exécutable
 - PIL-1.1.7.win32-py2.6.exe

12.3.4 Installer MySQL

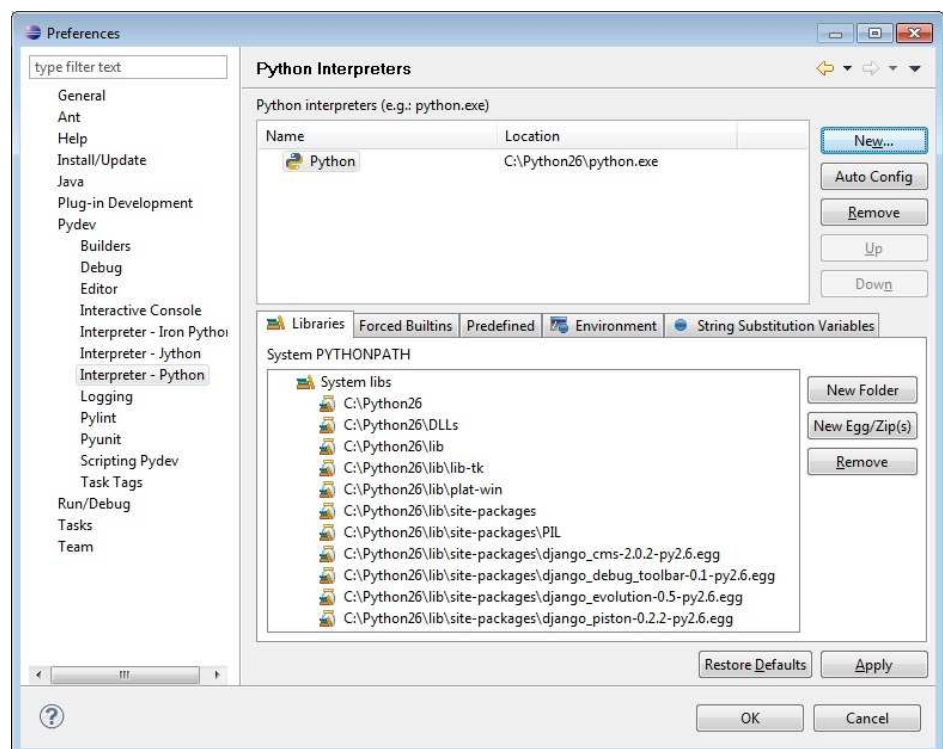
- ❑ mysql-essential-5.0.67-win32.msi
- ❑ mysql-gui-tools-5.0-r17-win32.msi (MySQL Administrator)
- ❑ MySQL-Python-1.2.3c1.win32-py2.6.exe (Plugin pour Django)

12.3.5 Installer la base de données

- ❑ Ouvrir MySQL Administrator
- ❑ A gauche, sélectionnez "Restore"
- ❑ Cliquez sur le bouton au fond à droite "Open backup"
- ❑ Sélectionnez le fichier "bdd.sql" dans le dossier "database" sur le CD-ROM
- ❑ Cliquez sur "Start restore"

12.3.6 Installation du plugin eclipse "Py-Dev"

- ❑ Ouvrir eclipse
- ❑ Aller dans le menu "Help" → "Install new software"
- ❑ Cliquer sur le bouton "Add" et insérer l'url suivante <http://pydev.org/updates>
- ❑ Cliquer sur le bouton "Next" puis "Finish"
- ❑ Ajouter le "Python path"
 - Aller dans le menu du sommet et cliquer "Window" → "Preferences" → "PyDev" → "Interpreter - Python"
 - Cliquer sur "New" et aller chercher l'exécutable (fichier *.exe) placé généralement dans "C:\Python26\Python.exe"
 - Cliquez sur "Ok" → "Apply" → "OK"



12.3.7 Ouvrir le projet Django

- ❑ Dans eclipse, faire "File" → "Import" → "Général" → "Existing project into workspace" → "Next"
- ❑ Cliquer sur "Browse" et aller sélectionner le dossier "Workspace" à la racine du CD-ROM.
- ❑ Le projet apparait dans la partie intermédiaire de la fenêtre. Sélectionnez-le et cliquez sur "Finish"
- ❑ Le projet peut maintenant être modifié et exécuté. Pour visualiser la plateforme. Ouvrir l'explorateur et tapez l'URL suivante : <http://localhost:8000>

12.4 Contenu du CD-ROM

A la racine du CD-ROM ci-joint se trouve un fichier nommé "index.html". A l'intérieur de ce dernier, l'arborescence est affichée afin de permettre une navigation facilitée. Voici une liste des différents documents présents sur ce CD-ROM :

- ❑ Les rapports hebdomadaires
- ❑ Le cahier des charges
- ❑ Le projet éclipse contenant le code source de l'application
- ❑ Le gadget Windows avec son code source
- ❑ La planification MS-Project avec le suivi
- ❑ Les processus des différents acteurs
- ❑ Les procès-verbaux des séances
- ❑ Les sources qui ont aidé à la réalisation du projet
- ❑ Les outils nécessaires à la mise en place de l'environnement
- ❑ Les prototypes écrans
- ❑ Une copie de la base de données

12.5 Détail des heures de travail

TOTAL PROJECT		363:00:00			
		<u>Matin</u>		<u>Après-midi</u>	
		Arrivée	Départ	Arrivée	Départ
lundi, 17. mai 2010	02:00	08:00	10:00		
mardi, 18. mai 2010	02:00			14:00	16:00
mercredi, 19. mai 2010	00:00				
jeudi, 20. mai 2010	05:30	08:30	11:30	12:30	15:00
vendredi, 21. mai 2010	02:00			12:00	14:00
samedi, 22. mai 2010	00:00				
dimanche, 23. mai 2010	00:00				
lundi, 24. mai 2010	00:00				
mardi, 25. mai 2010	03:30			13:00	16:30
mercredi, 26. mai 2010	02:30			14:00	16:30
jeudi, 27. mai 2010	07:15	08:30	12:00	12:45	16:30
vendredi, 28. mai 2010	02:00	08:30	10:30		
samedi, 29. mai 2010	00:00				
dimanche, 30. mai 2010	02:30			15:30	18:00
lundi, 31. mai 2010	09:00	08:30	11:30	13:00	19:00
mardi, 1. juin 2010	00:00				
mercredi, 2. juin 2010	00:00				
jeudi, 3. juin 2010	00:00				
vendredi, 4. juin 2010	00:00				
samedi, 5. juin 2010	00:00				
dimanche, 6. juin 2010	00:00				
lundi, 7. juin 2010	02:00	08:00	10:00		
mardi, 8. juin 2010	01:45			13:15	15:00

mercredi, 9. juin 2010	00:00				
jeudi, 10. juin 2010	07:30	08:00	12:00	14:00	17:30
vendredi, 11. juin 2010	03:00	10:00	13:00		
samedi, 12. juin 2010	00:00				
dimanche, 13. juin 2010	00:00				
lundi, 14. juin 2010	06:45	09:00	11:45	13:30	17:30
mardi, 15. juin 2010	06:45	09:00	11:15	13:00	17:30
mercredi, 16. juin 2010	00:00				
jeudi, 17. juin 2010	00:00				
vendredi, 18. juin 2010	03:00	09:30	11:30	13:00	14:00
samedi, 19. juin 2010	02:00	10:00	12:00		
dimanche, 20. juin 2010	00:00				
lundi, 21. juin 2010	00:00				
mardi, 22. juin 2010	04:30	10:00	12:00	13:30	16:00
mercredi, 23. juin 2010	00:00				
jeudi, 24. juin 2010	00:00				
vendredi, 25. juin 2010	00:00				
samedi, 26. juin 2010	00:00				
dimanche, 27. juin 2010	00:00				
lundi, 28. juin 2010	00:00				
mardi, 29. juin 2010	02:00	09:00	11:00		
mercredi, 30. juin 2010	00:00				
jeudi, 1. juillet 2010	04:30	15:00	18:00	20:00	21:30
vendredi, 2. juillet 2010	03:00	09:00	12:00		
samedi, 3. juillet 2010	00:00				
dimanche, 4. juillet 2010	04:00			14:00	18:00
lundi, 5. juillet 2010	08:15	08:30	12:00	13:15	18:00
mardi, 6. juillet 2010	07:40	08:40	12:45	14:00	17:35
mercredi, 7. juillet 2010	07:40	09:00	11:45	12:45	17:40
jeudi, 8. juillet 2010	07:45	09:00	13:45	15:00	18:00
vendredi, 9. juillet 2010	06:10	08:50	12:30	13:30	16:00
samedi, 10. juillet 2010	00:00				
dimanche, 11. juillet 2010	00:00				
lundi, 12. juillet 2010	03:30	08:30	12:00		
mardi, 13. juillet 2010	07:00	09:45	12:45	13:30	17:30
mercredi, 14. juillet 2010	08:05	08:25	11:50	13:05	17:45
jeudi, 15. juillet 2010	06:05	08:35	11:30	13:05	16:15
vendredi, 16. juillet 2010	07:30	08:30	11:30	12:30	17:00
samedi, 17. juillet 2010	00:00				
dimanche, 18. juillet 2010	00:00				
lundi, 19. juillet 2010	10:10	08:10	12:10	12:50	19:00
mardi, 20. juillet 2010	08:10	08:10	12:20	14:00	18:00
mercredi, 21. juillet 2010	08:30	08:20	12:20	13:00	17:30
jeudi, 22. juillet 2010	08:00	08:30	11:30	13:45	18:45

vendredi, 23. juillet 2010	07:40	08:35	11:50	12:50	17:15
samedi, 24. juillet 2010	00:00				
dimanche, 25. juillet 2010	07:00			12:40	19:40
lundi, 26. juillet 2010	11:10	08:35	11:50	12:50	20:45
mardi, 27. juillet 2010	09:30	08:30	17:00	20:00	21:00
mercredi, 28. juillet 2010	10:00	08:45	12:00	13:00	19:45
jeudi, 29. juillet 2010	08:00	10:30	11:45	13:15	20:00
vendredi, 30. juillet 2010	08:00	08:30	11:30	13:00	18:00
samedi, 31. juillet 2010	02:00	10:00	12:00		
dimanche, 1. août 2010	00:00				
lundi, 2. août 2010	07:00	10:00	12:00	13:45	18:45
mardi, 3. août 2010	12:00	08:00	12:00	13:00	21:00
mercredi, 4. août 2010	08:00	09:00	12:00	13:00	18:00
jeudi, 5. août 2010	09:00	09:00	12:00	13:30	19:30
vendredi, 6. août 2010	08:30	08:00	12:00	13:30	18:00
samedi, 7. août 2010	00:00				
dimanche, 8. août 2010	07:00	08:30	11:30	14:00	18:00
lundi, 9. août 2010	09:30	08:30	12:15	13:00	18:45
mardi, 10. août 2010	08:30	08:00	11:30	13:00	18:00
mercredi, 11. août 2010	08:40	08:20	12:00	13:00	18:00
jeudi, 12. août 2010	09:00	08:45	12:00	13:00	18:45
vendredi, 13. août 2010	08:30	08:30	12:00	13:00	18:00
samedi, 14. août 2010	08:00	08:30	11:30	12:00	17:00
dimanche, 15. août 2010	02:30	09:00	11:30		
lundi, 16. août 2010	00:00				